

# Learning Mixtures of Sparse Linear Regressions Using Sparse Graph Codes

Dong Yin<sup>1</sup>, Ramtin Pedarsani<sup>2</sup>, Yudong Chen<sup>3</sup>, and Kannan Ramchandran<sup>1</sup>

<sup>1</sup>Department of Electrical Engineering and Computer Sciences, UC Berkeley

<sup>2</sup>Department of Electrical and Computer Engineering, UC Santa Barbara

<sup>3</sup>School of Operations Research and Information Engineering, Cornell University

March 3, 2017

## Abstract

In this paper, we consider the *mixture of sparse linear regressions* model. Let  $\beta^{(1)}, \dots, \beta^{(L)} \in \mathbb{C}^n$  be  $L$  unknown sparse parameter vectors with a total of  $K$  non-zero coefficients. Noisy linear measurements are obtained in the form  $y_i = \mathbf{x}_i^H \beta^{(\ell_i)} + w_i$ , each of which is generated randomly from one of the sparse vectors with the label  $\ell_i$  unknown. The goal is to estimate the parameter vectors efficiently with low sample and computational costs. This problem presents significant challenges as one needs to simultaneously solve the *demixing* problem of recovering the labels  $\ell_i$  as well as the *estimation* problem of recovering the sparse vectors  $\beta^{(\ell)}$ .

Our solution to the problem leverages the connection between modern coding theory and statistical inference. We introduce a new algorithm, *Mixed-Coloring*, which samples the mixture strategically using query vectors  $\mathbf{x}_i$  constructed based on ideas from sparse graph codes. Our novel code design allows for both efficient demixing and parameter estimation. The algorithm achieves the order-optimal sample and time complexities of  $\Theta(K)$  in the noiseless setting, and near-optimal  $\Theta(K \text{ polylog}(n))$  complexities in the noisy setting. In one of our experiments, to recover a mixture of two regressions with dimension  $n = 500$  and sparsity  $K = 50$ , our algorithm is more than 300 times faster than EM algorithm, with about 1/3 of its sample cost.

## 1 Introduction

Mixture and latent variable models, such as Gaussian mixtures and subspace clustering, are expressive, flexible, and widely used in a broad range of problems including background modeling [1], speaker identification [2] and recommender systems [3]. However, parameter estimation in mixture models is notoriously difficult due to the non-convexity of the likelihood functions and the existence of local optima. In particular, it often requires a large sample size and many re-initializations of the algorithms to achieve an acceptable accuracy.

Our goal is to develop provably fast and efficient algorithms for mixture models — with sample and time complexities *sublinear* in the problem’s ambient dimension when the parameter vectors of interest is sparse — by leveraging the underlying low-dimensional structures.

In this paper we focus on a powerful class of models called *mixtures of linear regressions* [4]. We consider the *sparse* setting with a *query-based* algorithmic framework. In particular, we assume that each query-measurement pair  $(\mathbf{x}_i, y_i)$  is generated from a sparse linear model chosen randomly from  $L$  possible models:<sup>1</sup>

$$y_i = \mathbf{x}_i^H \beta^{(\ell)} + w_i \quad \text{with probability } q_\ell, \quad \text{for } \ell \in [L], \quad (1)$$

<sup>1</sup>We use  $\mathbf{x}_i^H$  to denote the conjugate transpose of  $\mathbf{x}_i$ , and  $[L]$  the set of integers  $\{1, 2, \dots, L\}$ .

where  $w_i$  is noise. The total number of nonzero elements in the parameter vectors  $\{\beta^{(\ell)} \in \mathbb{C}^n, \ell \in [L]\}$  is assumed to be  $K$ . The goal is to estimate the  $\beta^{(\ell)}$ 's, without knowing which  $\beta^{(\ell)}$  generates each query-measurement pair.

A mixture of regressions provides a flexible model for various heterogeneous settings where the regression coefficients differ for different subsets of observations. This model has been applied to a broad range of tasks including medicine measurement design [5], behavioral health care [6] and music perception modeling [7]. Here, we study the problem when the query vectors  $\mathbf{x}_i$  can be *designed* by the user; in Section 1.2 we discuss several practical applications that motivate the study of this query-based setting. Our results show that by appropriately exploiting this design freedom, one can achieve significant reduction the sample and computational costs.

To recover  $K$  unknown non-zero elements, it is clear that the amount of measurements and time required scale at least as  $\Theta(K)$ . We introduce a new algorithm, called the *Mixed-Coloring* algorithm, that *matches these sublinear sample and time complexity lower bounds*. The design of query vectors and decoding algorithm leverages ideas from sparse graph codes such as low-density parity-check (LDPC) codes [8]. Our algorithm recovers the parameter vectors with optimal  $\Theta(K)$  sample and time complexities in the noiseless setting, both in theory and empirically, and is stable under noise with near-optimal  $\Theta(K \text{ polylog}(n))$  sample and time complexities. Prior literature on this problem that does not utilize the design freedom typically have sample/time complexities that are at least polynomial in  $n$ ; we provide a survey of prior work and a more detailed comparison in Section 6. Empirically, we find that our algorithm is orders of magnitude faster than standard Expectation-Maximization (EM) algorithms for mixture of regressions. For example, in one of our experiments, detailed in Section 5, we consider recovering a mixture of two regressions with dimension  $n = 500$  and sparsity  $K = 50$ ; our algorithm is more than 300 times faster than EM algorithm, with about 1/3 of its sample cost.

## 1.1 Algorithm Overview

Our Mixed-Coloring algorithm solves two problems simultaneously: (i) rapiddemixing, namely identifying the *label*  $\ell_i$  of the vector  $\beta^{(\ell_i)}$  that generates each measurement  $y_i$ ; (ii) efficient identification of the *location* and *value* of the non-zero elements of the  $\beta^{(\ell)}$ 's. The main idea is to use a divide-and-conquer approach that iteratively reduce the original problem into simpler ones with much sparser parameter vectors. More specifically, we design  $\Theta(K)$  sets of sparse query vectors, with each set only associated with a subset of all the non-zero elements. The design of the query vectors ensures that we can first identify the sets which are associated with a single non-zero element (called singletons), and recover the location and value of that element (we call them singleton balls, shown as shaded balls in Figure 1b). We further identify the pairs of singleton balls which have the same (but unknown) label, indicated by the edges in Figure 1b. Results from random graph theory guarantees that, with high probability, the  $L$  largest connected components (giant components) of the singleton graph have the different labels, and thus we recover a fraction of the non-zero elements in each  $\beta^{(\ell)}$ , as shown in Figure 1c. We can then iteratively enlarge the recovered fraction with a guess-and-check method until finding all the non-zero elements. We revisit Figure 1 when describing the details of our algorithm in Section 3.

## 1.2 Motivation

Our problem is a natural extension of the setting of compressive sensing,<sup>2</sup> in which one often has full freedom of designing query vectors in order to estimate a sparse parameter vector. In many applications, the unknown sparse parameter vector can be affected by latent variables, leading to a mixture of sparse linear regressions, and these scenarios have been observed in neuroscience [9], genetics [10], psychology [5], etc. Here, we provide a concrete example motivated by neuroscience applications [9]. In neural signal processing, sensors are used to measure the brain activities, represented by an unknown sparse vector  $\beta$ . The sensors can be modeled as digital filters, and one can *design* the linear filter weights ( $\mathbf{x}_i$ 's) when measuring the neural

---

<sup>2</sup>Compressive sensing is a special case of our problem with  $L = 1$ .

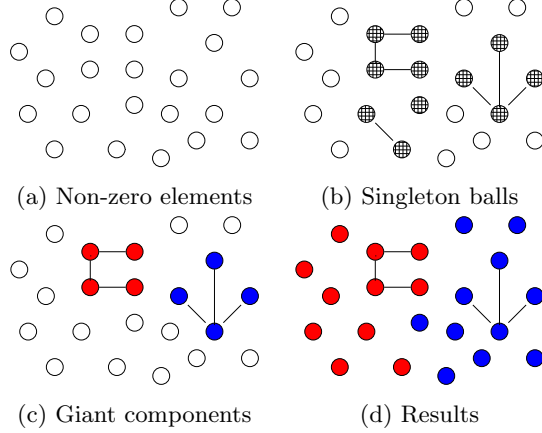


Figure 1: Mixed-Coloring algorithm with  $L = 2$ .

signal. Multiple sensors are usually placed in a particular area of the brain in order to acquire enough compressed measurements. However, there may be more than one neuron affecting a particular area of the brain, as shown in Figure 2, and each neuron may have different activities, corresponding to a different  $\beta^{(\ell)}$ . Consequently, each sensor may be measuring one of several different sparse signals, which can be formulated as a mixture-of-sparse-linear-regressions problem. Variants of this problem, such as neural spike sorting [9], has been studied in neuroscience. While the common solution is to use clustering algorithms on the spike signals, we believe that our algorithm provides the potential of improving sensor design and reducing sample and time complexities.

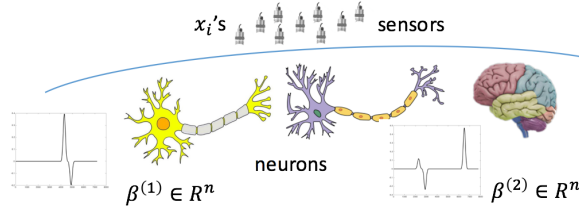


Figure 2: Mixture of neural signals.

In addition, our work adds the intellectual value of the power of design freedom in tackling sparse mixture problems by highlighting the huge performance gap between algorithms that can exploit the design freedom and those that cannot. We also believe that our ideas are applicable more broadly for other latent-variable problems that require experimental designs, such as survey designs in psychology with mixed type of respondents and biology experiments with mixed cell interior environments.

## 2 Main Results

In this section, we present the recovery guarantees for the Mixed-Coloring algorithm, and provide bounds on its sample and time complexities. We assume there are  $L$  unknown  $n$ -dimensional parameter vectors  $\beta^{(1)}, \dots, \beta^{(L)}$ . Each  $\beta^{(\ell)}$  has  $K_\ell$  non-zero elements, i.e.,  $|\text{supp}(\beta^{(\ell)})| = |\{j : \beta_j^{(\ell)} \neq 0\}| = K_\ell$ . Let  $K = \sum_{\ell=1}^L K_\ell$  be the total number of non-zero elements. Using the query vectors  $\{\mathbf{x}_i\} \in \mathbb{C}^n$ , the Mixed-Coloring algorithms obtains  $m$  measurements  $y_i$ ,  $i \in [m]$  generated independently according to the model (1), and outputs an estimate  $\{\hat{\beta}^{(\ell)}, \ell \in [L]\}$  of the unknown parameter vectors. We defer more details to Sections 3 and 4.

Our results are stated in the asymptotic regime where  $n$  and  $K$  approach infinity. A constant is a quantity that does not depend on  $n$  and  $K$ , with the associated Big-O notations  $\mathcal{O}(\cdot)$  and  $\Theta(\cdot)$ . We assume that  $L$  is a known and fixed constant, and the mixture weights satisfy  $q_\ell = \Theta(1)$  for each  $\ell \in [L]$  and thus are of the same order. Similarly, the sparsity levels of the parameter vectors are also of the same order with  $K_\ell = \Theta(K)$ .

## 2.1 Guarantees for the Noiseless Setting

In the noiseless case, i.e.,  $w_i \equiv 0$ , we consider for generality the complex-valued setting with  $\beta^{(\ell)} \in \mathbb{C}^n$  (our results can be easily applied to real case). We make a mild technical assumption, which stipulates that if any pair of parameter vectors have overlapping support, then the elements in the overlap are different.

**Assumption 1.** For each pair  $\ell_1, \ell_2 \in [L]$ ,  $\ell_1 \neq \ell_2$  and each index  $j \in \text{supp}(\beta^{(\ell_1)}) \cap \text{supp}(\beta^{(\ell_2)})$ , we have  $\beta_j^{(\ell_1)} \neq \beta_j^{(\ell_2)}$ .

Under the above setting, we have the following recovery guarantees for the Mixed-Coloring algorithm.

**Theorem 1.** Consider the asymptotic regime where  $n$  and  $K$  approach infinity. Under Assumption 1, for any fixed constant  $p^* \in (0, 1)$ , there exists a constant  $C > 0$  such that if the number of measurements is  $m = CK$ , then the Mixed-Coloring algorithm satisfies the following three properties for each  $\ell \in [L]$  (up to a label permutation):

1. (No False Discovery) For each  $j \in \text{supp}(\beta^{(\ell)})$ ,  $\hat{\beta}_j^{(\ell)}$  equals either  $\beta_j^{(\ell)}$  or 0; for each  $j \notin \text{supp}(\beta^{(\ell)})$ ,  $\hat{\beta}_j^{(\ell)} = 0$ .
2. (Element-wise Recovery) There exists a constant  $\tilde{p}_\ell \in (0, p^*)$  such that  $\mathbb{P}\{\hat{\beta}_j^{(\ell)} = \beta_j^{(\ell)}\} = 1 - \tilde{p}_\ell - \mathcal{O}(1/K)$  for each  $j \in \text{supp}(\beta^{(\ell)})$ .
3. (Support Recovery)

$$\mathbb{P}\{|\text{supp}(\hat{\beta}^{(\ell)})| \geq (1 - p^*)|\text{supp}(\beta^{(\ell)})|\} = 1 - \mathcal{O}(1/K).$$

Moreover, the computational time of the Mixed-Coloring algorithm is  $\Theta(K)$ .

The theorem ensures that the Mixed-Coloring algorithm has no false discovery, and recovers  $(1 - p^*)$  fraction of the non-zero elements with high probability. The error fraction  $p^*$  is an input parameter to algorithm, and can be made arbitrarily close to zero by adjusting the oversampling ratio  $C \equiv C(p^*, L, \{q_\ell\})$ . (By more careful analysis, one can show that the dependence of  $C$  on  $p^*$  is  $C = \mathcal{O}(\log(1/p^*))$ . Here, since we set  $p^*$  as a constant,  $C$  is a constant.) Given the number of components  $L$ , mixture weights  $\{q_\ell\}$  and the target  $p^*$ , the value of the constant  $C$  can be computed numerically. The table below gives some of the  $C$  values for several  $p^*$  and  $L$ , under the setting  $q_\ell = 1/L, \forall \ell \in [L]$ . We see that the value of  $C$  is quite modest.

Table 1: Sample complexity of the Mixed-Coloring algorithm

$L$	2	3	4
$p^*$	$5.1 \times 10^{-6}$	$8.8 \times 10^{-6}$	$8.1 \times 10^{-6}$
$m = CK$	$33.39K$	$37.80K$	$40.32K$

We can in fact boost the above guarantee to recover all the non-zero elements, by running the Mixed-Coloring algorithm  $\Theta(\log K)$  times independently and aggregating the results by majority voting. By property 2 in Theorem 1 and a union bound argument, this procedure *exactly* recovers all the parameter vectors with probability  $1 - \mathcal{O}(1/\text{poly}(K))$  with  $\Theta(K \log K)$  sample and time complexities.

## 2.2 Guarantees for the Noisy Setting

An extension of the previous algorithm, *Robust Mixed-Coloring*, handles noise in the measurement model (1). Here we focus on the case with two parameter vectors which appear equally likely, i.e.,  $L = 2$  and  $q_\ell = 1/2$ ,  $\ell = 1, 2$ . Many interesting applications have binary latent factors: gene mutation present/not, gender, healthy/sick individual, children/adult, etc. The noise  $w_i$  is assumed to be i.i.d. Gaussian with mean zero and constant variance  $\sigma^2$ . For the purpose of theoretical analysis, we assume that the non-zero elements in the parameter vectors take value in a finite quantized set.

**Assumption 2.** *The non-zero elements of the parameter vectors satisfy  $\beta_j^{(\ell)} \in \mathbb{D}, \forall \beta_j^{(\ell)} \neq 0, \ell \in [L]$ , where*

$$\mathbb{D} \triangleq \{\pm\Delta, \pm 2\Delta, \dots, \pm b\Delta\} \subset \mathbb{R},$$

*The positive constants  $\Delta$  and  $b$  are known to the algorithms.*

As shown in our empirical results in Section 5, the Robust Mixed-Coloring algorithm works even when the assumption is violated. In this case, the algorithm produces the best quantized approximation to the unknown parameter vectors, provided that they are not too far off the quantized set. The theoretical results for the continuous alphabet setting is still an open problem, and the tools in recent work such as [11] may be applied to our problem.

When the quantization assumption holds, exact recovery is possible, as guaranteed in the theorem below. The Robust Mixed Coloring algorithm maintains sublinear sample and time complexities, and recovers the parameter vectors in the presence of noise with bounded variance.

**Theorem 2.** *Consider the asymptotic regime where  $K$  and  $n$  approach infinity with  $K = \Theta(n^\alpha)$  for some constant  $\alpha \in (0, 1]$ . When  $L = 2$  and Assumptions 1 and 2 hold, there exists a constant  $\eta > 0$ , such that if  $\Delta/\sigma > \eta$  and the number of measurements is  $m = \Theta(K \text{polylog}(n))$ , then the Robust Mixed-Coloring algorithm satisfies the three properties in Theorem 1. Moreover, the computational time of the Robust Mixed-Coloring algorithm is  $\Theta(K \text{polylog}(n))$ .*

Similar to the noiseless case, by running the Robust Mixed-Coloring algorithm  $\Theta(\log K)$  times, one can exactly recover the two parameter vectors with probability  $1 - \mathcal{O}(1/\text{poly}(K))$ . In this case, the sample and computational complexities are  $\Theta(K \log(K) \text{polylog}(n))$ , and further, since we assume that  $K = \Theta(n^\alpha)$  for some constant  $\alpha$ , we can still conclude that the sample and computational complexities for full recovery are  $\Theta(K \text{polylog}(n))$ .

## 3 Mixed-Coloring Algorithm for Noiseless Recovery

In this section, we provide details of the Mixed-Coloring algorithm in the noiseless setting. We first provide some primitives that serve as important ingredients in the algorithm, and then describe the design of query vectors and decoding algorithm in detail.

### 3.1 Primitives

The algorithm makes uses of four basic primitives: **summation check**, **indexing**, **peeling**, and **guess-and-check**, which are described below.

**Summation Check:** Suppose that we generate two query vectors  $\mathbf{x}_1$  and  $\mathbf{x}_2$  independently from some continuous distribution on  $\mathbb{C}^n$ , and a third query vector of the form  $\mathbf{x}_1 + \mathbf{x}_2$ . Let  $y_1$ ,  $y_2$ , and  $y_3$  be the corresponding measurements. We check the sum of the measurements and in the noiseless case, if  $y_3 = y_1 + y_2$ , then with probability one, we know that these three measurements are generated from the same parameter vector  $\beta^{(\ell)}$ . In this case we call  $\{y_1, y_2\}$  a *consistent pair* of measurements as they are from the same  $\beta^{(\ell)}$  (the third measurement  $y_3$  is now redundant).

**Indexing:** The indexing procedure is to find the locations and values of the non-zero elements by carefully designed query vectors. In the noiseless case, this can be done by suitably designed *ratio test*. We sketch the idea of the ratio test here. Consider a consistent pair of measurements  $\{y_1, y_2\}$  and corresponding query vectors  $\{\mathbf{x}_1, \mathbf{x}_2\}$ . We design the query vectors such that the information of the locations of the non-zero elements is encoded in the relative phase between  $y_1$  and  $y_2$ . In particular, we generate  $n$  i.i.d. random variables  $r_j, j \in [n]$  uniformly distributed on the unit circle. Letting  $W = e^{i\frac{2\pi}{n}}$  where  $i$  is the imaginary unit, we set the  $j$ -th entries of  $\mathbf{x}_1$  and  $\mathbf{x}_2$  to be either  $x_{1,j} = x_{2,j} = 0$ , or  $x_{1,j} = r_j$  and  $x_{2,j} = r_j W^{j-1}$ . (The locations of the zeros are determined using sparse-graph codes and discussed later.) Below is an example of such a consistent pair of measurements and the corresponding linear system:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1^H \\ \mathbf{x}_2^H \end{bmatrix} \boldsymbol{\beta}^{(1)} = \begin{bmatrix} 0 & r_2 & r_3 & 0 & 0 & r_6 & 0 & 0 \\ 0 & r_2 W & r_3 W^2 & 0 & 0 & r_6 W^5 & 0 & 0 \end{bmatrix} \boldsymbol{\beta}^{(1)}. \quad (2)$$

Suppose that  $\boldsymbol{\beta}^{(1)}$  is 3-sparse and of the form  $\boldsymbol{\beta}^{(1)} = [0 \ 0 \ * \ 0 \ 0 \ 0 \ *]^T$ . There is only one non-zero element,  $\beta_3^{(1)}$ , that contributes to the measurements  $y_1$  and  $y_2$ . In this case the consistent measurement pair  $\{y_1, y_2\}$  is called a *singleton*. A singleton can be detected by testing the integrality of the relative phase of the ratio  $y_1/y_2$ . In the above example, since  $y_1 = r_3 \beta_3^{(1)}$  and  $y_2 = r_3 W^2 \beta_3^{(1)}$ , we observe that  $|y_1| = |y_2|$  and the relative phase  $\angle(y_2/y_1) = 2 \cdot \frac{2\pi}{8}$  is an integral multiple of  $\frac{2\pi}{8}$ . We therefore know that with probability one, this consistent pair is a singleton, and moreover the corresponding non-zero element is located at the 3-rd coordinate with value  $\beta_3^{(1)} = y_1/r_3$ . We would like to remark that the indexing step can also be done using real-valued query vectors.

**Peeling:** The third ingredient of the decoder is peeling, i.e., iteratively reducing the problem by subtracting off recovered elements, in a Gaussian elimination-like manner. In the example above, suppose instead that  $\boldsymbol{\beta}^{(1)}$  is 4-sparse, i.e.,  $\boldsymbol{\beta}^{(1)} = [0 \ * \ * \ 0 \ * \ 0 \ 0 \ *]^T$ , in which case the consistent pair

$$y_i = x_{i,2} \beta_2^{(1)} + x_{i,3} \beta_3^{(1)}, \quad i = 1, 2 \quad (3)$$

is associated with two non-zero elements of  $\boldsymbol{\beta}^{(1)}$ . If in a previous iteration of the algorithm we have recovered the location and value of  $\beta_2^{(1)}$ , then we can subtract/peel off this recovered element by  $y_i \leftarrow y_i - x_{i,2} \beta_2^{(1)}$ , for  $i = 1, 2$ .

The updated measurement pairs satisfy  $y_i = x_{i,3} \beta_3^{(1)}, i = 1, 2$ , and we have reduced the problem to a simpler form. In fact, in this case the pair  $\{y_1, y_2\}$  becomes a singleton, to which the above ratio test can be applied to recover  $\beta_3^{(1)}$ .

**Guess-and-check:** The ratio test and peeling steps can be combined to detect that two non-zero elements are from the same parameter vectors. In the previous example (3), suppose instead that we recovered two elements  $\beta_2^{(\ell_1)}$  and  $\beta_3^{(\ell_2)}$  in previous iterations via ratio-testing another two consistent pairs that are singletons, but values of their labels  $\ell_1$  and  $\ell_2$  are unknown. We can still try to peel off  $\beta_2^{(\ell_1)}$  from  $\{y_1, y_2\}$ ; if the updated measurements  $\{y_1, y_2\}$  pass the ratio test and recover a non-zero element with location 3 and value  $\beta_3^{(\ell_2)}$ , then we know that with probability one the non-zero elements  $\beta_2^{(\ell_1)}$  and  $\beta_3^{(\ell_2)}$  must come from the same parameter vector (the one that generates  $\{y_1, y_2\}$ ), i.e.,  $\ell_1 = \ell_2 = 1$ . In this case the peeling step is valid.

The continuing execution of these four primitives is made possible by the design of the query vectors using sparse-graph codes, which we describe next.

### 3.2 Design of Query Vectors

As illustrated in Figure 3, we construct  $M = \Theta(K)$  sets of query vectors (called *bins*). The query vectors in each bin are associated with some coordinates of the parameter vectors (i.e., the queries are non-zero only on those coordinates). The association between the coordinates and bins is determined by a  $d$ -left regular bipartite graph with  $n$  left nodes (coordinates) and  $M$  right nodes (bins), where each left node is connected to  $d = \Theta(1)$  right nodes chosen independently uniformly at random. Each bin consists of three query vectors.

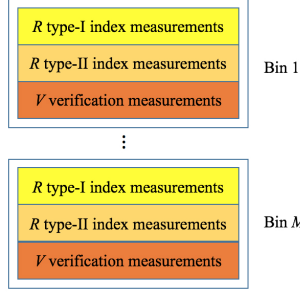


Figure 3:  $(2R + V)M$  query vectors.

The values of the non-zero elements of the first two query vectors are in the form of (2), enabling the ratio test. The third query vectors equals the sum of the first two and is used for the summation check.

If the query vectors in each bin were used only once, then we would have very few bins passing the summation check and hence few consistent pairs. Instead, we use the first two query vectors repeatedly for  $R = \Theta(1)$  times, obtaining two sets of measurements, each of size  $R$  and called *type-I* and *type-II index measurements*. We use the third query vector  $V = \Theta(1)$  times to obtain a set of *verification measurements*. We therefore have  $2R + V$  measurements associated with each of the  $M$  bins, hence a total of  $m = (2R + V)M = \Theta(K)$  measurements, as shown in Figure 3. Using density evolution methods [12], we can find proper values of  $d$ ,  $R$ ,  $V$ , and  $M$  such that successful recovery is guaranteed.

### 3.3 Decoding Algorithm

The decoding algorithm first finds consistent pairs (by summation check) in each bin, within which singletons are identified (by the ratio test). The ratio test also recovers the location and values of several non-zero elements, some of which can then be associated with the same  $\beta^{(\ell)}$  by guess-and-check. At this point, for each  $\beta^{(\ell)}$ , we have recovered some of its non-zero elements (including their locations, values and labels). These steps are then repeated iteratively via peeling until no more non-zero elements can be found. Below we elaborate on these steps.

**Finding Consistent Pairs:** The decoding procedure starts by finding all the consistent pairs. In each bin, we perform summation checks on all triplets  $(y_1, y_2, y_3)$  in which  $y_1$ ,  $y_2$ , and  $y_3$  are the type-I index measurement, type-II index measurement and verification measurement, respectively. If a triplet passes the summation check, then a consistent pair  $\{y_1, y_2\}$  is found. Note that in each bin the number of triplets of the above form is a constant, so this step can be done in  $\Theta(K)$  time. The subsequent steps of the algorithm are based on the consistent pairs found in this step.

**Recovering a Subset of Non-zero Elements:** Each non-zero element of the parameter vectors can be identified by its label-location-value triplet  $(\ell, j, \beta_j^{(\ell)})$ . We visualize these triplets (i.e., non-zero elements) as balls, as shown in Figure 1a, and initially their labels, locations and values are unknown. As before, a consistent pair associated with only one non-zero element is called a singleton, and we call this non-zero element a *singleton ball*. We run the ratio test on the consistent pairs to identify singletons and their associated singleton balls. The singleton balls found are illustrated in Figure 1b as shaded balls. The ratio test also recovers the locations and values of these singleton balls, although at this point we do not know the label  $\ell$  of the balls.

The next step is crucial: For two singleton balls and a consistent measurement pair associated with the locations of these two balls, we run the guess-and-check operations to detect if these two singleton balls indeed have the same label (or equivalently, if the two non-zero elements are in the same parameter vector). If so, we connect these two balls with an edge, as shown in Figure 1b. Doing so creates a graph over the

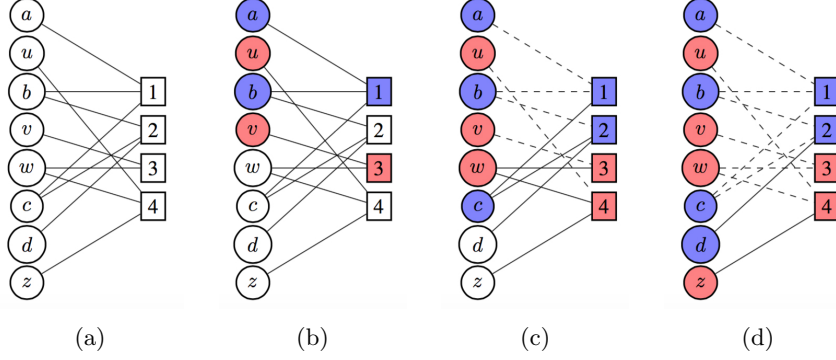


Figure 4: Iterative decoding. If a ball is peeled off, the edges connected to it are shown in dashed lines. The colored balls in (b) are found by the giant component method. In (c) and (d), more balls are colored by iterative decoding.

balls (i.e., non-zero elements), and each connected component of the graph is from a single parameter vector. Since each non-zero element is associated with a constant number of consistent pairs (due to using a  $d$ -left regular bipartite graph with constant  $d$ ), this step can in fact be done efficiently in  $\Theta(K)$  time without enumerating all the combinations of singleton ball pairs.

By carefully choosing the parameters  $d$ ,  $M$ ,  $R$ , and  $V$ , and using tools from random graph theory, we can ensure that with high probability the  $L$  largest connected components (called *giant components*) correspond to the  $L$  parameter vectors, and each of these components has size  $\Theta(K)$ . Then, the labels of the balls in these components are now identified. This is illustrated in Figure 1c for  $L = 2$ , where colors represent the labels. In summary, at this point we have recovered the labels, locations and values of a constant fraction of the non-zero elements (i.e., balls) of each parameter vector.

**Iterative Decoding:** The decoding procedure proceeds by identifying the labels of the remaining balls via iteratively applying the peeling and guess-and-check primitives. The connected components in Figure 1c are therefore expanded, until no more changes can be made, as illustrated in Figure 1d.

We provide an example of this iterative procedure in Figure 4. Recall that the association between the coordinates of the parameter vectors and the bins (or consistent pairs) is determined by a bipartite graph. Here, we only show one consistent pair for each bin and omit the zero elements. The non-zero elements and the consistent pairs are shown as balls and squares, respectively, as in Figure 4a. The steps described in the last part recover a subset of these balls, which are shown in colors in Figure 4b. Now consider the measurement pair 1, which is associated with the balls  $a$ ,  $b$  and  $c$ . As  $a$  and  $b$  are recovered, we can peel them off from the measurement pair 1 to recover (by the ratio test) the label, location and value of the non-zero element represented by ball  $c$ . Similarly, peeling off the recovered ball  $v$  from the measurement pair 3, recovers ball  $w$ , as illustrated in Figure 4c. We continue this process iteratively, peeling off balls recovered in the previous iterations to recover more balls. For example, we peel off the balls  $b$  and  $c$  from the measurement pair 2 to recover the ball  $d$ , and the ball  $w$  from pair 4 to recover ball  $z$ , resulting in Figure 4d. So far we have described the Mixed-Coloring algorithm in the noiseless case, and we refer readers to Section B of the appendices for the analysis of this algorithm.

## 4 Robust Mixed-Coloring Algorithm for Noisy Recovery

The overall structure of the Robust Mixed-Coloring algorithm is the same as its noiseless counterpart. In the presence of noise, the ratio test method for indexing and the summation check primitive need to be robustified, which are done by a modification of the query design. In particular, we design three types of query vectors. The first type, called *binary indexing* vectors, encodes the location information using binary representations with,  $\lceil \log_2(n) \rceil$  bits (as opposed to using the relative phases in the noiseless case). A similar



approach is considered in [13] for compressive phase retrieval. The second type is called *singleton verification* vectors, which are used for singleton detection. Using these two types of vectors we can modify the ratio test to achieve the same performance with noise. The third type of query vectors is used for *consecutive summation check*, which finds *consistent sets* of measurements.

In addition to the new query design, we also employ a noise reduction scheme. This is done by using each designed query vector (say  $\mathbf{x}_i$ ) repeatedly for  $R$  times and averaging the corresponding measurements from the same  $\beta^{(\ell)}$ . In particular, these  $R$  measurements are sampled i.i.d. from a mixture of two Gaussians with centers  $\mathbf{x}_i^T \beta^{(1)}$  and  $\mathbf{x}_i^T \beta^{(2)}$ , so we use an EM algorithm initialized by moment methods to estimate the two centers. Using the result in [14], we prove that the EM-based noise reduction scheme succeeds under the conditions in Theorem 2, namely  $R = \Theta(\text{polylog}(n))$  and  $\Delta/\sigma > \eta$ . We refer the readers to Section D of the appendices for the details of the Robust Mixed-Coloring algorithm, and Section F for more details of the EM algorithm that we use.

## 5 Experimental Results

In this section, we test the sample and time complexities of the Mixed-Coloring algorithm in both noiseless and noisy cases to verify our theoretical results. We refer the readers to the appendices for more details of the experiments.

For the noiseless case, we use the optimal parameters  $(d, R, V)$  from numerical calculations of the density evolution. For different values of  $L, K, m$ , we record the empirical success probability and running time averaged over 100 trials. Here, we use a sufficiently small  $p^*$  so that the success event is equivalent to recovery of *all* the non-zero elements. The results are shown in Figure 5a. The phase transition occurs at some  $C = m/K$  that matches the values in Table 1 predicted by our theory. Moreover, the running time is linear in  $K$  and does not depend on  $n$ , as shown in Figure 5b.

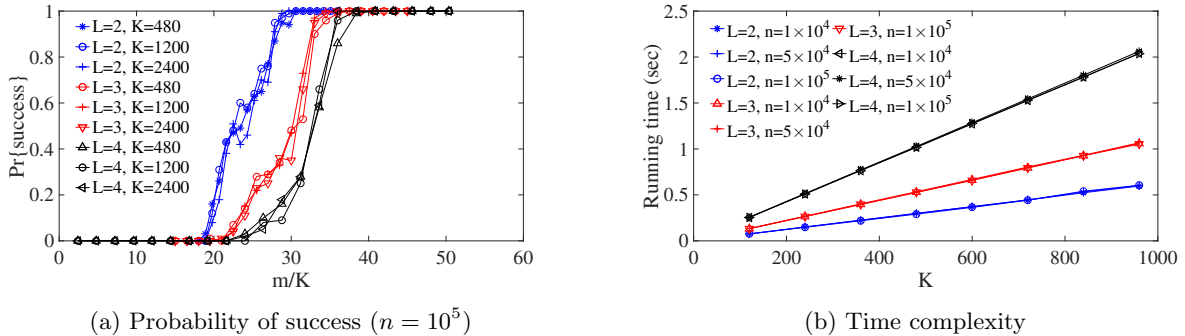


Figure 5: Success probability and running time in the noiseless case.

Similar experiments are performed for the noisy case using the Robust Mixed-Coloring algorithm, under the quantization assumption. Figure 6a shows the minimum number of queries  $m$  required for 100 consecutive successes, for different  $n$  and  $K$ . We observe that the sample complexity is linear in  $K$  and sublinear in  $n$ . The running time exhibits a similar behavior, as shown in Figure 6b. Both observations agree with the prediction of our theory.

We also compare the Mixed-Coloring algorithm with a state-of-the-art EM-style algorithm (equivalent to alternating minimization in the noiseless setting) from [15]. These comparisons are not entirely fair, since our algorithm is based on carefully designed query vectors, while the algorithm in [15] uses random design, i.e., the entries of  $\mathbf{x}_i$ 's are i.i.d. Gaussian. However, this is exactly where the intellectual value of our work lies: we expose the gains available by careful design. We consider four test cases with  $(L, n, K) = (2, 100, 20), (2, 500, 50), (2, 100, 100), (2, 500, 500)$ , with the first two cases being sparse problems and the last two being relatively dense problems. We find the minimum number of queries that leads to a 100% successful rate in 100 trials, and the average running time. As shown in Table 2, in both sparse and dense problems, our

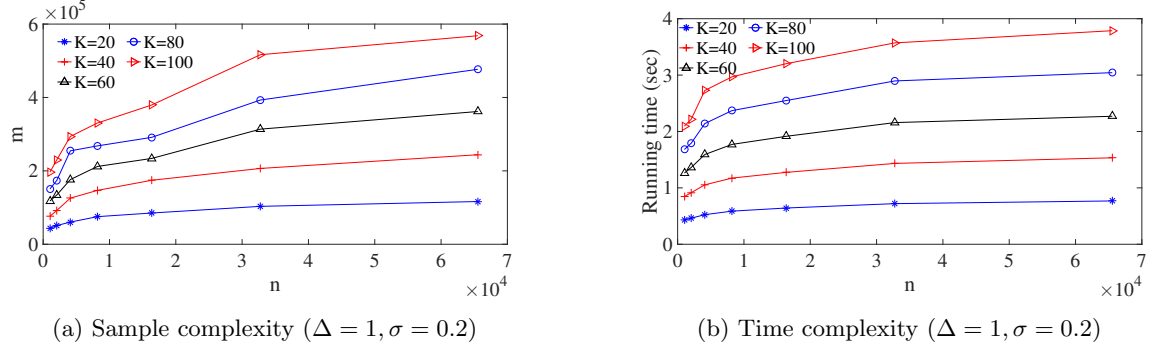


Figure 6: Sample and time complexities of Robust Mixed-Coloring algorithm.

Mixed-Coloring algorithm is several orders of magnitude faster. As for the sample complexity, our algorithm requires smaller number of samples in the sparse cases, while in dense problems, the sample complexity of our algorithm is within a constant factor (about 3) of that of the alternating minimization algorithm. For the noisy setting, our algorithm is most powerful in the high dimensional setting, i.e., large  $n$ , due to the  $\text{polylog}(n)$  factors. However, in this setting, it takes extremely long time for the state-of-the-art algorithms such as [16] to converge, and thus, we do not present the comparison in the noisy setting.

Table 2: Comparison of two algorithms  
(M-C=Mixed-Coloring)

$(n, K)$	$\frac{\text{sample(M-C)}}{\text{sample(EM)}}$	$\frac{\text{speed(M-C)}}{\text{speed(EM)}}$
(100, 20)	0.57	124
(500, 50)	0.33	368
(100, 100)	2.78	19
(500, 500)	3.00	37

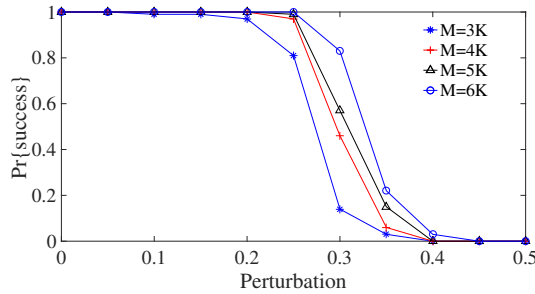


Figure 7: Performance of Robust Mixed-Coloring algorithm with quantization assumption violated.

We further test the Robust Mixed-Coloring algorithm when the quantization assumption is violated. For any  $\beta \in \mathbb{R}$ , we define  $D(\beta) = \arg \min_{a \in \mathbb{D}} |a - \beta| \mathbf{1}(\beta \neq 0)$ , where  $\mathbf{1}(\cdot)$  denotes the indicator function. This means that  $D(\beta)$  is the element in  $\mathbb{D}$  which is the closest one to  $\beta$ , when  $\beta \neq 0$ . For a vector  $\beta \in \mathbb{R}^n$ , we define  $D(\beta) = \{D(\beta_j)\}_{j=1}^n$ . We define the *perturbation* of a vector  $\beta$  as  $\text{Perturbation}(\beta) = \max_{j \in [n]} |\beta_j - D(\beta_j)|/\Delta$ .

In this experiment, we generate sparse parameter vectors  $\beta^{(\ell)}$ ,  $\ell \in [L]$  with a total number of  $K$  non-zero elements. These non-zero elements are generated randomly while keeping the perturbation of the parameter vectors under a certain level by adding bounded noise to the quantized non-zero elements. We record the probability of success for different number of bins  $M$  and different perturbation level. Here the success event

is defined as recovery of  $D(\beta^{(\ell)})$  for all  $\ell \in [L]$ . The result is shown in Figure 7. We see that the Robust Mixed-Coloring algorithm works without the quantization assumption as long as the perturbations are not too large.

## 6 Related Work

### 6.1 Mixtures of Regressions

Parameter estimation using the expectation-maximization (EM) algorithm is studied empirically in [17]. In [16], an  $\ell_1$ -penalized EM algorithm is proposed for the sparse setting. Theoretical analysis of the EM algorithm is difficult due to non-convexity. Progress was made in [15] and [14] under stylized Gaussian settings with dense  $\beta$ , for which a sample complexity of  $\Theta(n \text{polylog}(n))$  is proved given a suitable initialization of EM. The algorithm uses a grid search initialization step to guarantee that the EM algorithm can find the global optimal solution, with the assumption that the query vectors are i.i.d. Gaussian distributed. The computational complexity is polynomial of  $n$ . An alternative algorithm is proposed in [18], which achieves optimal  $\mathcal{O}(n)$  sample complexity, but has high computational cost due to the use of semidefinite lifting. The algorithm in [19] makes use of tensor decomposing techniques, but suffers from a high sample complexity of  $\mathcal{O}(n^6)$ . In comparison, our approach has order optimal sample and time complexities by utilizing the potential design freedom.

### 6.2 Coding-theoretic Methods

Many modern error-correcting codes such as LDPC codes and polar codes [20] with their roots in communication problems, exploit redundancy to achieve robustness, and use structural design to allow for fast decoding. These properties of codes have recently found applications in statistical problems, including graph sketching [21], sparse covariance estimation [22], low-rank approximation [23], and discrete inference [24]. Most related to our approach is the work in [25, 26, 13], which apply sparse graph codes with peeling-style decoding algorithms to compressive sensing and phase retrieval problems. In our setting we need to handle a mixture distribution, which requires more sophisticated query design and novel unmixing algorithms that go beyond the standard peeling-style decoding.

### 6.3 Combinatorial and Dimension Reduction Techniques

Our results demonstrate the power of strategic query and coding theoretic tools in mixture problems, and can be considered as efficient linear sketching of a mixture of sparse vectors. In this sense, our work is in line with recent work that make uses of combinatorial and dimension reduction techniques in high-dimensional and large scale statistical problems. These techniques, such as locality-sensitive hashing [27], sketching of convex optimization [28], and coding-theoretic methods [29], allow one to design highly efficient and robust algorithms applicable to computationally challenging datasets without compromising statistical accuracy.

## 7 Conclusions

We propose the Mixed-Coloring algorithm as a query based learning algorithm for mixtures of sparse linear regressions. The design of the query vectors and the recovery algorithm are base sparse graph codes, and our scheme achieves order optimal sample and computational complexities in the noiseless case, and sublinear sample and time complexities in the presence of noise. Our experiments justified the theoretical results. In the noisy scenario, studying the Robust Mixed-Coloring algorithm with more than two parameter vectors and obtain theoretical results for the continuous alphabet can be two important future directions.

## References

- [1] M. Harville, “A framework for high-level feedback to adaptive, per-pixel, mixture-of-gaussian background models,” in *Computer Vision ECCV 2002*. Springer, 2002, pp. 543–560.
- [2] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, “Speaker verification using adapted gaussian mixture models,” *Digital signal processing*, vol. 10, no. 1, pp. 19–41, 2000.
- [3] A. Zhang, N. Fawaz, S. Ioannidis, and A. Montanari, “Guess who rated this movie: Identifying users through subspace clustering,” *arXiv preprint arXiv:1208.1544*, 2012.
- [4] R. De Veaux, “Mixtures of linear regressions,” *Comp. Statistics & Data Analysis*, vol. 8, no. 3, 1989.
- [5] E. Blackwell, C. F. M. de Leon, and G. E. Miller, “Applying mixed regression models to the analysis of repeated-measures data in psychosomatic medicine,” *Psychosomatic Medicine*, vol. 68, no. 6, 2006.
- [6] P. Deb and M. Holmes, “Estimates of use and costs of behavioural health care: a comparison of standard and finite mixture models,” *Econometric Analysis of Health Data*, pp. 87–99, 2002.
- [7] K. Viele and B. Tong, “Modeling with mixtures of linear regressions,” *Statistics and Computing*, vol. 12, no. 4, pp. 315–330, 2002.
- [8] R. Gallager, “Low-density parity-check codes,” *IRE Transactions on information theory*, vol. 8, no. 1, pp. 21–28, 1962.
- [9] M. S. Lewicki, “A review of methods for spike sorting: the detection and classification of neural action potentials,” *Network: Computation in Neural Systems*, vol. 9, no. 4, pp. R53–R78, 1998.
- [10] R. Jansen, “A general mixture model for mapping quantitative trait loci by using molecular markers,” *Theoretical and Applied Genetics*, vol. 85, no. 2-3, pp. 252–260, 1992.
- [11] D. Yin, R. Pedarsani, X. Li, and K. Ramchandran, “Compressed sensing using sparse-graph codes for the continuous-alphabet setting,” *54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2016.
- [12] T. Richardson and R. Urbanke, “The capacity of low-density parity-check codes under message-passing decoding,” *IEEE Transactions on Information Theory*, vol. 47, pp. 599–618, February 2001.
- [13] D. Yin, K. Lee, R. Pedarsani, and K. Ramchandran, “Fast and robust compressive phase retrieval with sparse-graph codes,” in *IEEE International Symposium on Information Theory*, 2015, pp. 2583–2587.
- [14] S. Balakrishnan, M. J. Wainwright, and B. Yu, “Statistical guarantees for the em algorithm: From population to sample-based analysis,” *arXiv preprint:1408.2156*, 2014.
- [15] X. Yi, C. Caramanis, and S. Sanghavi, “Alternating minimization for mixed linear regression,” in *Proceedings of The 31st International Conference on Machine Learning*, 2014, pp. 613–621.
- [16] N. Städler, P. Bühlmann, and S. Van De Geer, “ $\ell_1$ -penalization for mixture regression models,” *Test*, vol. 19, no. 2, pp. 209–256, 2010.
- [17] S. Faria and G. Soromenho, “Fitting mixtures of linear regressions,” *Journal of Statistical Computation and Simulation*, vol. 80, no. 2, pp. 201–225, 2010.
- [18] Y. Chen, X. Yi, and C. Caramanis, “A convex formulation for mixed regression with two components: Minimax optimal rates,” *arXiv preprint arXiv:1312.7006*, 2013.
- [19] A. T. Chaganty and P. Liang, “Spectral experts for estimating mixtures of linear regressions,” in *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, 2013, pp. 1040–1048.

- [20] E. Arıkan, “Channel polarization a method for constructing capacity-achieving codes for symmetric binary-input memoryless channels,” *IEEE Transactions on Information Theory*, vol. 55, no. 7, 2009.
- [21] X. Li and K. Ramchandran, “An active learning framework using sparse-graph codes for sparse polynomials and graph sketching,” in *Advances in Neural Information Processing Systems*, 2015, pp. 2161–2169.
- [22] R. Pedarsani, K. Lee, and K. Ramchandran, “Sparse covariance estimation based on sparse-graph codes,” in *Annual Allerton Conference on Communication, Control, and Computing*, 2015.
- [23] S. Ubaru, A. Mazumdar, and Y. Saad, “Low rank approximation using error correcting coding matrices,” in *Proceedings of the 32nd International Conference on Machine Learning*, 2015, pp. 702–710.
- [24] S. Ermon, C. Gomes, A. Sabharwal, and B. Selman, “Low-density parity constraints for hashing-based discrete integration,” in *Proc. 31st International Conference on Machine Learning*, 2014, pp. 271–279.
- [25] X. Li, S. Pawar, and K. Ramchandran, “Sub-linear time support recovery for compressed sensing using sparse-graph codes,” *arXiv preprint arXiv:1412.7646*, 2014.
- [26] R. Pedarsani, K. Lee, and K. Ramchandran, “Phasecode: Fast and efficient compressive phase retrieval based on sparse-graph-codes,” *arXiv preprint arXiv:1408.0034*, 2014.
- [27] I. S. Dhillon, P. K. Ravikumar, and A. Tewari, “Nearest neighbor based greedy coordinate descent,” in *Advances in Neural Information Processing Systems*, 2011, pp. 2160–2168.
- [28] M. Pilanci and M. J. Wainwright, “Iterative hessian sketch: Fast and accurate solution approximation for constrained least-squares,” *arXiv preprint arXiv:1411.0347*, 2014.
- [29] D. Achlioptas and P. Jiang, “Stochastic integration via error-correcting codes,” in *Proc. Uncertainty in Artificial Intelligence*, 2015.
- [30] W. B. Johnson and J. Lindenstrauss, “Extensions of lipschitz mappings into a hilbert space,” *Contemporary mathematics*, vol. 26, no. 189-206, p. 1, 1984.

# Appendices

## A Details of Experiments

In this section, we provide more details of the experiments that we conducted. All simulations are done on a laptop with 2.8 GHz Intel Core i7 CPU and 16 GB memory using Python.

In Figure 5, we test the success probability and running time in the noiseless case. In both Figure 5a and Figure 5b, we use  $(d, R, V) = (15, 3, 3)$  for  $L = 2$ ,  $(d, R, V) = (15, 5, 5)$  for  $L = 3$ ,  $(d, R, V) = (13, 8, 8)$  for  $L = 4$ . In Figure 5b, we use  $M/K = 3.8$  for  $L = 2$ ,  $M/K = 2.6$  for  $L = 3$ ,  $M/K = 1.8$  for  $L = 4$ .

In Table 2, we compare the sample and time complexities of the Mixed-Coloring algorithm and the alternating minimization algorithm. We use  $d = 15$ ,  $R = V = 3$  and  $M = 3.8K$ . The parameters for alternating minimization are chosen as suggested in the original paper [15].

In Figure 6, we test the sample and time complexities of the Robust Mixed-Coloring algorithm. In both Figure 6a and Figure 6b, we choose quantization level  $\Delta = 1$ , standard deviation of noise  $\sigma = 0.2$ , algorithm parameters:  $d = 15$ ,  $M = 3K$ , number of singleton verification query vectors:  $0.3 \log_2(n)$ . In Figure 6a, we vary  $R$  to find the minimum number of query vectors needed for successful recovery. In Figure 6b, we fix  $R = \log_2(n)$  and test the time cost.

In Figure 7, we test the performance of Robust Mixed-Coloring algorithm with quantization assumption violated. We vary the number of bins  $M$  to test the empirical probability of success, and also keep  $d = 5M/K$ . Other parameters:  $n = 4096$ ,  $K = 50$ , quantization level  $\Delta = 1$ , standard deviation of noise  $\sigma = 0.1$ , number of singleton verification query vectors:  $0.3 \log_2(n)$ , and  $R = \log_2(n)$ .

## B Proof of Theorem 1

### B.1 Proof Outline

We prove Theorem 1 in this section. The proof includes two major steps: (i) show that the expectation of the fraction of non-zero elements which are not recovered can be arbitrarily small; (ii) show that this fraction concentrates around its mean with high probability. The first part mainly uses density evolution techniques which is commonly used in coding theory, and the second part uses Doob's martingale argument.

### B.2 Notations

We briefly recall the Mixed-Coloring algorithm in the noiseless case and declare some notations that we will use for the rest of the proof.

Recall that the parameter vector  $\beta^{(\ell)}$  has  $K_\ell$  non-zero elements. We call these  $K_\ell$  non-zero elements *balls* in *color*  $\ell$ . We design a  $d$ -left regular bipartite graph with  $n$  left nodes and  $M$  right nodes, representing the  $n$  coordinates and the  $M$  bins, respectively. We denote the  $i$ -th bin by  $\mathcal{B}_i$ . We use the matrix  $\mathbf{H} \in \{0, 1\}^{M \times n}$  to represent the biadjacency matrix of the bipartite graph, i.e.,  $H_{i,j} = 1$  if and only if the  $i$ -th bin is associated with the  $j$ -th coordinate. Recall that we design three query vectors for in the form of (2), for the purpose of ratio test. The third query vectors is the summation of the first two and is used for summation check. We repeat the first two query vectors  $R$  times, respectively, and get  $R$  type-I and  $R$  type-II index measurements. We repeat the third query vector  $V$  times and get  $V$  verification measurements. For the  $j$ -th verification measurement of the  $i$ -th bin, we define a *sub-bin*  $\mathcal{B}_i^j$ . If we can find one type-I index measurement and one type-II index measurement such that the summation of the two measurements is equal to the  $j$ -th verification measurement, we know that these three measurements are generated by the same parameter vector, say  $\beta^{(\ell)}$ . The two index measurements are called a consistent pair. Then, we say that the sub-bin  $\mathcal{B}_i^j$  has color  $\ell$ . We define the *color set*  $\mathcal{C}_i^j$  of  $\mathcal{B}_i^j$ . If we can find a consistent pair corresponding to the  $j$ -th verification measurement, we let  $\mathcal{C}_i^j = \{\ell\}$ , otherwise  $\mathcal{C}_i^j = \emptyset$ . We further define the color set of bin  $\mathcal{B}_i$  as  $\mathcal{C}_i = \cup_{j=1}^V \mathcal{C}_i^j$ .

### B.3 Number of Singleton Balls

In this section, we analyze the number of singleton balls in color  $\ell$  found in the first stage of the algorithm. We can show that this number is concentrated around a constant fraction of  $K_\ell$  with high probability.

**Lemma 1.** *Let  $K_s^{(\ell)}$  be the number of singleton balls in color  $\ell$  found in the first stage. Then, there exists a constant<sup>3</sup>  $q_s^{(\ell)}$  such that for any constant  $\delta > 0$ ,*

$$\mathbb{P}\{|K_s^{(\ell)} - K_\ell q_s^{(\ell)}| \leq \delta K_\ell\} \geq 1 - 2\exp(-2\delta^2 K_\ell). \quad (4)$$

*Proof.* We first specify some terminologies here. For a bin  $\mathcal{B}_i$ , we say that this bin *has* color  $\ell$  when  $\ell \in \mathcal{C}_i$ . One should notice that if there are more than one sub-bins in color  $\ell$  in bin  $\mathcal{B}_i$ , these sub-bins are identical, and therefore, we can say that a bin  $\mathcal{B}_i$  *contains*  $k$  balls in color  $\ell$ , when  $\mathcal{B}_i$  has at least one sub-bin  $\mathcal{B}_i^j$  in color  $\ell$ , and the sub-bin is associated with  $k$  non-zero elements in  $\beta^{(\ell)}$ , or equivalently, the coded parameter vector  $\tilde{\beta}_i^j = \text{diag}(\mathbf{h}_i^T) \beta^{(\ell)}$  satisfies  $|\text{supp}(\tilde{\beta}_i^j)| = k$ ,  $k \geq 0$ .

First, we analyze the probability  $Q_\ell$  that a particular bin  $\mathcal{B}_i$  has color  $\ell$ . According to our model, the measurements are generated independently, therefore, we have

$$Q_\ell = [1 - (1 - q_\ell)^V][1 - (1 - q_\ell)^R]^2.$$

Then, we use  $\xi_k^{(\ell)}$  to denote the probability of the event that a particular bin contains  $k$  balls in color  $\ell$ . Since each ball is associated with  $d$  bins among the  $M$  bins independently and uniformly at random, the number of balls in color  $\ell$  that a bin contains is binomial distributed with parameters  $K_\ell$  and  $\frac{d}{M}$ , and we have

$$\xi_k^{(\ell)} = Q_\ell \binom{K_\ell}{k} \left(\frac{d}{M}\right)^k \left(1 - \frac{d}{M}\right)^{K_\ell - k}.$$

In addition, we can use Poisson distribution to approximate the binomial distribution when  $\lambda_\ell := \frac{K_\ell d}{M}$  is a constant and  $K_\ell$  approaches infinity. In the following analysis, we will use the approximation

$$\xi_k^{(\ell)} \approx Q_\ell \frac{\lambda_\ell^k e^{-\lambda_\ell}}{k!}.$$

Consider the bipartite graph representing the association between the balls in color  $\ell$  and the  $M$  bins. We know that there are  $K_\ell d$  edges connected to the balls in color  $\ell$ , and we use  $\rho_k^{(\ell)}$  to denote the expected fraction of these  $K_\ell d$  edges which are connected to a bin which contains  $k$  balls in color  $\ell$ ,  $k \geq 1$ . Then, we have

$$\rho_k^{(\ell)} = \frac{kM}{K_\ell d} \xi_k^{(\ell)} = Q_\ell \frac{\lambda_\ell^{k-1} e^{-\lambda_\ell}}{(k-1)!},$$

and equivalently,  $\rho_k^{(\ell)}$  is also the probability that an edge, which is chosen from the  $K_\ell d$  edges uniformly at random, is connected to a bin  $\mathcal{B}_i$  containing  $k$  balls in color  $\ell$ .

Let  $q_s^{(\ell)}$  be the probability that a ball in color  $\ell$  is a singleton ball. The event that this ball is a singleton ball is equivalent to the event that at least one of its  $d$  associated bins contains one ball color  $\ell$ . Then, when  $K_\ell$  approaches infinity, we have

$$q_s^{(\ell)} = 1 - (1 - \rho_1^{(\ell)})^d,$$

and this is because in the limit  $K_\ell \rightarrow \infty$ , the correlations between the  $d$  edges connected to a ball become negligible; this technique is often used in the theoretical analysis of density evolution in coding theory, and we will use this type of asymptotic argument several times in the proofs. Let  $K_s^{(\ell)}$  be the number of singleton balls in color  $\ell$ , then we have  $\mathbb{E}[K_s^{(\ell)}] = K_\ell q_s^{(\ell)}$ . Using the asymptotic argument and by Hoeffding's inequality, we also have for any constant  $\delta > 0$ ,

$$\mathbb{P}\{|K_s^{(\ell)} - K_\ell q_s^{(\ell)}| \leq \delta K_\ell\} \geq 1 - 2\exp(-2\delta^2 K_\ell),$$

and this means that the number of singleton balls in color  $\ell$  is highly concentrated around  $K_\ell q_s^{(\ell)}$ . □

<sup>3</sup>Recall that in our paper, constants are defined as quantities which do not depend on  $n$  and  $K$ .

## B.4 Initial Fractions

We construct the graph  $\mathcal{G}_\ell$  whose nodes correspond to the singleton balls in color  $\ell$  found in the previous stage, and analyze the number of edges in  $\mathcal{G}_\ell$ , which is equal to the number of strong doubletons in color  $\ell$ . Then, we can show that the number of strong doubletons is concentrated around a constant fraction of  $M$  with high probability.

**Lemma 2.** *Let  $M_s^{(\ell)}$  be the number of strong doubletons in color  $\ell$  found in the second stage. Then, there exists a constant  $\nu_\ell > 0$  such that for any constant  $\delta > 0$ ,*

$$\mathbb{P}\{|M_s^{(\ell)} - M\nu_\ell| \leq \delta M\} \geq 1 - 2\exp(-2\delta^2 M). \quad (5)$$

*Proof.* We know that the expected number of doubletons in color  $\ell$  is  $M\xi_2^{(\ell)}$ . Then, we analyze the probability that a doubleton is a strong doubleton. Similar to the analysis in [26], for a particular ball in color  $\ell$ , we let  $B$  denote the event that this ball is in a singleton, and  $D$  denote the event that this ball is in a doubleton. We have the conditional probability that a ball in a doubleton is also a singleton ball:

$$\begin{aligned} q_d^{(\ell)} &:= \mathbb{P}\{B|D\} = \frac{\mathbb{P}\{D \cap B\}}{\mathbb{P}\{D\}} \\ &= \frac{1 - \mathbb{P}\{\bar{B}\} - \mathbb{P}\{\bar{D}\} + \mathbb{P}\{\bar{B} \cap \bar{D}\}}{1 - \mathbb{P}\{\bar{D}\}} \\ &= \frac{1 - (1 - \rho_1^{(\ell)})^d - (1 - \rho_2^{(\ell)})^d + (1 - \rho_1^{(\ell)} - \rho_2^{(\ell)})^d}{1 - (1 - \rho_2^{(\ell)})^d}. \end{aligned}$$

Then we know the probability that a doubleton is a strong doubleton is  $(q_d^{(\ell)})^2$ , and the expected number of strong doubletons in color  $\ell$  is  $M\xi_2^{(\ell)}(q_d^{(\ell)})^2$ . Let  $\nu_\ell = \xi_2^{(\ell)}(q_d^{(\ell)})^2$  and  $M_s^{(\ell)}$  be the number of edges in graph  $\mathcal{G}_\ell$ . The expectation of  $M_s^{(\ell)}$  is  $\mathbb{E}[M_s^{(\ell)}] = M\nu_\ell$ , and according to Hoeffding's inequality, we have for any  $\delta > 0$

$$\mathbb{P}\{|M_s^{(\ell)} - M\nu_\ell| \leq \delta M\} \geq 1 - 2\exp(-2\delta^2 M),$$

meaning that the number of edges is highly concentrated around  $M\nu_\ell$ .  $\square$

Then, we get the following result on the size of the giant component of  $\mathcal{G}_\ell$ , using the asymptotic behavior of the Erdos-Renyi random graphs.

**Lemma 3.** *Let  $K_G^{(\ell)}$  be the size of the largest connected component (giant component) of  $\mathcal{G}_\ell$ . If the parameters of the Mixed-Coloring algorithm satisfy*

$$\frac{2M\nu_\ell}{K_\ell q_s^{(\ell)}} > 1, \quad (6)$$

*then, for any constant  $\delta > 0$ , with probability  $1 - \mathcal{O}(1/K_\ell)$ , initial fraction of the balls in color  $\ell$  which are recovered after the second stage satisfies*

$$\left| \frac{K_G^{(\ell)}}{K_\ell} - \zeta_\ell q_s^{(\ell)} \right| \leq \delta, \quad (7)$$

*where the constant  $\zeta_\ell$  is the unique solution of the equation*

$$\zeta_\ell + \exp\left(-2\frac{\zeta_\ell M\nu_\ell}{K_\ell q_s^{(\ell)}}\right) = 1,$$

*and other connected components in  $\mathcal{G}_\ell$  are of sizes  $\mathcal{O}(\log(K_\ell))$ .*



*Proof.* This result is a direct corollary of the asymptotic behavior of the Erdos-Renyi random graphs, and we only give a brief proof here. First, we condition on the number of singleton balls that we find in the first stage, i.e.,  $K_s^{(\ell)}$  and the number of edges in  $\mathcal{G}_\ell$ , i.e.,  $M_s^{(\ell)}$ . By symmetry, we know that the  $M_s^{(\ell)}$  edges are uniformly chosen from the  $\binom{K_s^{(\ell)}}{2}$  possible edges. Therefore, the graph  $\mathcal{G}_\ell$  is an Erdos-Renyi random graph. According to the results on the giant component of Erdos-Renyi random graphs, we know that if the limit

$$\theta := \lim_{K_s^{(\ell)} \rightarrow \infty} K_s^{(\ell)} \frac{M_s^{(\ell)}}{\binom{K_s^{(\ell)}}{2}} > 1,$$

then with probability  $1 - \mathcal{O}(1/K_s^{(\ell)})$ , the size of the giant component of graph  $\mathcal{G}_\ell$  is linear in  $K_s^{(\ell)}$ , and other connected components have sizes  $\mathcal{O}(\log(K_s^{(\ell)}))$ . By (4) and (5), we know that for any constant  $\delta > 0$ , the limit  $\theta$  lies in the interval  $[(1 - \delta) \frac{2M\nu_\ell}{K_\ell q_s^{(\ell)}}, (1 + \delta) \frac{2M\nu_\ell}{K_\ell q_s^{(\ell)}}]$ , with probability  $1 - \mathcal{O}(\exp(-\alpha K_\ell))$ , for some constant  $\alpha > 0$ . Then, we can get rid of the conditioning and complete the proof of Lemma 3.  $\square$

## B.5 Tree-like Assumption

By Lemma 3, we know that we can recover a constant fraction of the non-zero elements with probability  $1 - \mathcal{O}(1/K_\ell)$ . Then, we study the iterative decoding process. The analysis is based on density evolution, which is a common and powerful technique in coding theory. Similar to other coding-theoretic analysis, our derivation of density evolution is based on a tree-like assumption. Here, we state the tree-like assumption first and provide the results on the probability that the tree-like assumption holds.

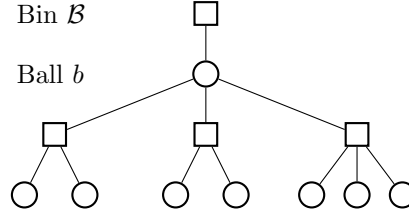


Figure 8: Level-2 neighborhood of edge  $(b, \mathcal{B})$ .

As we have mentioned, the association between the balls in color  $\ell$  (non-zero elements in  $\beta^{(\ell)}$ ) and the bins can be represented by a  $d$ -left regular bipartite graph. We label the edges by an ordered pair of a ball  $b$  and a bin  $\mathcal{B}$ , denoted by  $e = (b, \mathcal{B})$ . We define the *level- $C^*$*  neighborhood of  $e$ , denoted by  $N_e^{C^*}$  as the subgraph of all the edges and nodes on paths with length less than or equal to  $C^*$ , which start from  $b$  and the first edge of the paths are not  $e$  [26]. We have the following results on the probability that  $N_e^{C^*}$  is a tree, or equivalently, cycle-free, for a constant  $C^*$ .

**Lemma 4.** [26] *For a fixed constant  $C^*$ ,  $N_e^{2C^*}$  is a tree with probability at least  $1 - \mathcal{O}(\log(K_\ell)^{C^*}/K_\ell)$ .*

We conduct the density evolution analysis conditioned on the event that  $N_e^{2C^*}$  is a tree for an edge  $e$  which is chosen from the  $K_\ell d$  edges uniformly at random. Then, we will take the complementary event into consideration and complete the analysis.

## B.6 Analysis on the Density Evolution

Recall that in the first iteration, we find all the singletons, and in the second iteration, we find the strong doubletons and form the giant component. Let  $p_j^{(\ell)}$  be the probability that at the  $j$ th iteration of the learning algorithm, a ball in color  $\ell$ , which is chosen from the  $K_\ell$  balls uniformly at random, is not recovered,  $j \geq 2$ . Here,  $p_2^{(\ell)}$  corresponds to the probability that after the second iteration, a randomly chosen ball in color  $\ell$  is not in the giant component. According to the previous section, we know that by choosing parameters which

satisfy (6), we have  $p_2^{(\ell)} = \frac{K_\ell - K_G^{(\ell)}}{K_\ell} = \Theta(1)$  with probability  $1 - \mathcal{O}(1/K_\ell)$ . Now we analyze the relationship between  $p_{j+1}^{(\ell)}$  and  $p_j^{(\ell)}$  for  $j \geq 2$ .

Consider the iterative decoding process as a *message passing* process. First, we know that at iteration  $j + 1$ , a ball in color  $\ell$  passes a message to a bin through an edge claiming that it is colored, if and only if at least one of the other  $d - 1$  neighborhood bins contains a resolvable multiton in color  $\ell$ . Second, a sub-bin in color  $\ell$  becomes a resolvable multiton if and only if all the other balls in this sub-bin are colored. This message passing process is illustrated in Figure 8. Under the tree-like assumption, the messages passed among the balls and bins are independent, we have

$$p_{j+1}^{(\ell)} = (1 - \sum_{i=2}^{\infty} \rho_i^{(\ell)} (1 - p_j^{(\ell)})^{i-1})^{d-1},$$

which gives us

$$p_{j+1}^{(\ell)} = (1 - Q_\ell(e^{-\lambda_\ell p_j^{(\ell)}} - e^{-\lambda_\ell}))^{d-1}. \quad (8)$$

As we can see, the major difference between the density evolution of the Mixed-Coloring algorithm and the PhaseCode algorithm is that there is a constant probability  $Q_\ell$  that a bin has a sub-bin in color  $\ell$ .

Next, we will show that after a constant number of iterations,  $p_j^{(\ell)}$  can be arbitrarily small.

**Lemma 5.** *If we choose parameters satisfying*

$$(d - 1)Q_\ell \lambda_\ell e^{-\lambda_\ell t} > 1, \quad (9)$$

*then for any constant  $\delta > 0$ , there exists a constant  $T$ , such that  $p_T^{(\ell)} < \delta$ .*

*Proof.* Let  $f_\ell(t) = (1 - Q_\ell(e^{-\lambda_\ell t} - e^{-\lambda_\ell}))^{d-1}$ , then we have  $p_{j+1}^{(\ell)} = f_\ell(p_j^{(\ell)})$ . It is easy to see that  $f_\ell(1) = 1$ ,  $f_\ell(0) > 0$ , and  $f_\ell$  is a monotonically increasing function. We also have

$$f'_\ell(t) = (d - 1)Q_\ell \lambda_\ell e^{-\lambda_\ell t} (1 - Q_\ell(e^{-\lambda_\ell t} - e^{-\lambda_\ell}))^{d-2}.$$

We know that if there is

$$f'_\ell(1) = (d - 1)Q_\ell \lambda_\ell e^{-\lambda_\ell} > 1, \quad (10)$$

then there exists at least one fixed point  $t \in (0, 1)$  such that  $f_\ell(t) = t$ . We use  $p_\ell^*$  to represent the largest fixed point of  $f_\ell(t)$  in  $(0, 1)$ . Now we argue that the fixed point can be made arbitrarily small by choosing proper parameters. Suppose that for a certain set of parameters  $\lambda_\ell$  and  $d$ , the fixed point is  $p_\ell^*$ , then if we keep  $\lambda_\ell$  and increase  $d$  to  $\tilde{C}d$ , where  $\tilde{C} > 1$  is a constant, then we can see that the new fixed point is upper bounded by  $(p_\ell^*)^{\tilde{C}}$ , and in this way, the fixed point can be made an arbitrarily small constant. As shown in [26], as long as we can choose parameters to make the fixed point  $p_\ell^* < \delta/2$ , then, there exists a constant number of iterations  $T$ , depending on  $\delta$ , such that  $p_T^{(\ell)} < \delta$ .  $\square$

Then, we can prove the following lemma showing that the number of uncolored balls in color  $\ell$  is concentrated around  $K_\ell p_T^{(\ell)}$  with high probability.

**Lemma 6.** *Let  $Z_\ell$  be the number of uncolored balls in color  $\ell$  after  $T$  iterations. Then for any  $\delta > 0$ , there exists constant  $c_1$ , such that when conditioned on the event that  $p_2^{(\ell)} = \Theta(1)$ , and  $K_\ell$  is large enough,*

$$\left| \mathbb{E}[Z_\ell] - K_\ell p_T^{(\ell)} \right| < K_\ell \delta / 2, \quad (11)$$

$$\mathbb{P} \left\{ \left| Z_\ell - K_\ell p_T^{(\ell)} \right| > K_\ell \delta \right\} < 2 \exp \{ -c_1 \delta^2 K_\ell^{1/(4T+1)} \}. \quad (12)$$

The proof of Lemma 6 is the same as in [26], and uses Doob's martingale argument and Azuma's concentration bound. We should also notice that the event that the tree-like assumption does not hold is already considered in (11). Now combining Lemmas 3, 5, and 6, we have shown that for a specific color  $\ell$ , there exists proper parameters of the algorithm such that after a constant number of iterations, the Mixed-Coloring algorithm can recover an arbitrarily large fraction of the balls in color  $\ell$  with probability  $1 - \mathcal{O}(1/K_\ell)$ . Using a union bound over all the  $L$  colors ( $L$  is a constant), we have proved the results in Theorem 1 on the error probability.

## B.7 Computational Complexity

In this part, we analyze the computational complexity of the algorithm. First, since there are  $M = \Theta(K)$  bins and each bin has a constant number of sub-bins. Refining the measurements of each bin takes  $\Theta(1)$  operations, the computational complexity of refining measurements is  $\Theta(K)$ . Next, to find all the singletons, we need to check all the colored sub-bins, and checking each sub-bin takes  $\Theta(1)$  operations, the computational complexity of this stage is  $\Theta(K)$ . In the third stage, we find all the strong doubletons. We know that there are  $\Theta(K)$  singleton balls and for each singleton ball, there are  $d$  bins connected to it. For each of the bins, we subtract the measurements contributed by the singleton ball from the refined measurements in the sub-bins, and do the ratio test to see if it is a strong doubleton. Therefore, processing each bin takes  $\Theta(1)$  operations and since  $d$  is also a constant, the computational complexity of finding strong doubletons is also  $\Theta(K)$ . Then, we get the graph with  $\Theta(K)$  nodes and  $\Theta(K)$  edges, corresponding to the singleton balls and strong doubletons, respectively. Using breadth-first search algorithm, the computational complexity of finding the connected components is  $\Theta(K)$ . In the last stage, we iteratively find other uncolored balls. For each unprocessed sub-bin, since we do not know the color of the sub-bin, there are  $L$  possible remaining measurements. Each time when we find a new ball, we update at most  $dV$  remaining measurements and do the ratio test. Therefore, it takes  $\Theta(1)$  operations when coloring a new ball. Since there are  $\Theta(K)$  uncolored balls after finding the giant components, the computational complexity of the last stage is also  $\Theta(K)$ . So far, we have shown that the computational complexity of Mixed-Coloring algorithm is  $\Theta(K)$ , which completes the proof of Theorem 1.

## C Computing the Constants in the Sample Complexity

In this section, we give exact constants in the sample complexity results. For simplicity, we assume that  $K_\ell = K/L$  and  $q_\ell = 1/L$  for all  $\ell \in [L]$ . We define a new notation  $c = M/K$ , and then there is  $\lambda_\ell = \frac{d}{Lc}$ . We will analyze the minimum number of measurements that we need to reach a certain reliability target. More precisely, we set the maximum error floor to be  $p_{\max}^*$ , and numerically calculate the error floor for different values of  $d$ ,  $c$ ,  $R$ , and  $V$ . Then, we minimize the number of total measurements, which is proportional to  $(2R + V)c$  with the constraint that the error floor  $p^* \leq p_{\max}^*$ . As we have shown in previous parts, the parameters should also satisfy (6) and (9). We know that if (6) is satisfied, when  $K$  is large enough, there should be a giant component with size linear in  $K$  for each color. where  $\theta > 1$  is a threshold that we can choose. Therefore, we select parameters with three constraints, which are (9), (6), and  $p^* \leq p_{\max}^*$ .

The results of the numerical calculation are shown in Table 3. In these experiments, we set  $p_{\max}^* = 10^{-5}$ ,  $\theta = 2$ , and we fix the left degree  $d$  and choose different values of  $c$ ,  $R$ , and  $V$  to minimize the number of measurements with the three constraints. Then we compare the optimal number of measurements over different choices of  $d$  and find the optimal  $d$ . As we can see, to reach the same reliability level, for  $L = 2, 3, 4$ , the optimal number of measurements we need is  $33.39K$ ,  $37.80K$ , and  $40.32K$ , respectively. The number of measurements we need only increases slightly with  $L$ , and the optimal  $d$  is around 13 and 15.

## D Details of Noisy Recovery Algorithm

In this section, we provide more details to show we robustify the Mixed-coloring algorithm in presence of noise. The overall structure of the algorithm is the same as the noiseless case. However, one can see that the ratio test method that we use for indexing in the noiseless case and the summation check approach are both fragile to noise. Therefore, we need different design of query vectors. The main idea to robustify the algorithm is to encode the location information using binary representations, i.e.,  $\lceil \log(n) \rceil$  binary bits, rather than the relative phases. Similar methods have been used in problems such as compressive phase retrieval [13]. Further, instead of consistent pairs, we find *consistent sets* of measurements using *consecutive summation check*.

Table 3: Constants in the results of sample complexity.

$L = 2$	$d$	11	12	13	14	<b>15</b>	16	17	18
	$p^*/10^{-6}$	6.7	8.7	1.9	3.1	<b>5.1</b>	1.6	0.5	7.4
	$M/K$	2.95	3.17	3.23	3.46	<b>3.71</b>	3.78	3.86	4.37
	$R$	4	4	4	3	<b>3</b>	3	3	3
	$V$	4	3	3	4	<b>3</b>	3	3	2
	$m/K$	35.4	34.87	35.53	34.6	<b>33.39</b>	34.02	34.74	34.96
$L = 3$	$d$	11	12	13	14	<b>15</b>	16	17	18
	$p^*/10^{-6}$	4.4	5.2	2.7	9.2	<b>8.8</b>	2.8	6.2	2.3
	$M/K$	1.94	2.08	2.17	2.39	<b>2.52</b>	2.56	2.76	2.81
	$R$	7	6	6	5	<b>5</b>	5	5	5
	$V$	7	7	6	6	<b>5</b>	5	4	4
	$m/K$	40.74	39.52	39.06	38.24	<b>37.80</b>	38.4	38.64	39.34
$L = 4$	$d$	11	12	<b>13</b>	14	15	16	17	18
	$p^*/10^{-6}$	7.8	8.7	<b>8.1</b>	5.6	4.2	3.3	4.0	5.0
	$M/K$	1.48	1.59	<b>1.68</b>	1.76	1.85	1.93	2.04	2.16
	$R$	9	9	<b>8</b>	8	7	7	7	6
	$V$	11	8	<b>8</b>	7	8	7	6	7
	$m/K$	42.92	41.34	<b>40.32</b>	40.48	40.7	40.53	40.8	41.04

**Design of Queries.** We still design the query vectors according to the  $d$ -left regular bipartite graph. For a particular bin, let  $\mathbf{h} \in \{0, 1\}^n$  denote the association between this bin and the coordinates. We design  $P = \Theta(\log^2(n))$  query vectors  $\mathbf{x}_i \in \mathbb{R}^n$ ,  $i \in [P]$  for this bin as follows:

$$[\mathbf{x}_1 \ \cdots \ \mathbf{x}_P]^T = [\mathbf{B}^T \ \mathbf{V}^T \ \mathbf{C}^T]^T \text{diag}(\mathbf{h}).$$

where  $\mathbf{B} \in \{0, 1\}^{P_1 \times n}$ ,  $\mathbf{V} \in \{1, -1\}^{P_2 \times n}$ , and  $\mathbf{C} \in \mathbb{Z}^{P_3 \times n}$  are used for binary indexing, singleton verification, and summation check, respectively. The matrix  $\mathbf{B}$  has  $P_1 = \lceil \log_2(n) \rceil$  rows, and the  $j$ -th column of  $\mathbf{B}$  is the binary representation of integer  $j - 1$ . The matrix  $\mathbf{V}$  has  $P_2 = \Theta(\log(n))$  rows and consists of i.i.d. Rademacher entries, i.e., the entries of  $\mathbf{V}$  are equally likely to be either 1 or  $-1$ . The matrix  $\mathbf{C}$  contains  $P_3 = \binom{P_1 + P_2}{2}$  rows, and the rows of  $\mathbf{C}$  are indexed by pairs  $(i, k)$ ,  $1 \leq i < k \leq P_1 + P_2$ . Let  $\mathbf{D} = [\mathbf{B}^T \ \mathbf{V}^T]^T$  be a collection of the first two matrices. The row of  $\mathbf{C}$  indexed by  $(i, k)$  (denoted by  $\mathbf{c}_{(i,k)}^T$ ) is the summation of the  $i$ -th and the  $k$ -th row of  $\mathbf{D}$ , i.e.,  $\mathbf{c}_{(i,k)}^T = \mathbf{d}_i^T + \mathbf{d}_k^T$ . Here, we give a simple example for  $n = 4$  and  $P_2 = 2$ . We can see that we need  $P = 10$  query vectors in this case, as shown below:

$$[\mathbf{x}_1 \ \cdots \ \mathbf{x}_P]^T = \begin{bmatrix} \mathbf{b}_1^T \\ \mathbf{b}_2^T \\ \mathbf{v}_3^T \\ \mathbf{v}_4^T \\ \mathbf{c}_{(1,2)}^T \\ \mathbf{c}_{(1,3)}^T \\ \mathbf{c}_{(1,4)}^T \\ \mathbf{c}_{(2,3)}^T \\ \mathbf{c}_{(2,4)}^T \\ \mathbf{c}_{(3,4)}^T \end{bmatrix} \text{diag}(\mathbf{h}) = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ -1 & 1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 0 & 1 & 1 & 2 \\ -1 & 1 & 2 & 0 \\ 1 & 1 & 0 & 0 \\ -1 & 2 & 1 & 0 \\ 1 & 2 & -1 & 0 \\ 0 & 2 & 0 & -2 \end{bmatrix} \text{diag}(\mathbf{h}).$$

Now we can describe how the indexing process works. The indexing process consists of three ingredients, *noise reduction*, *consecutive summation check*, and *singleton detection*. We elaborate these three ingredients as below.

**Noise Reduction.** Due to the presence of noise, the first step that we need to take is a noise reduction operation. Specifically, we use each query vector  $T = \Theta(\text{polylog}(n))$  times, repeatedly. According to our

model of mixed linear regressions, in the presence of Gaussian noise, one can see that if  $\mathbf{x}_i^T \boldsymbol{\beta}^{(1)} = \mathbf{x}_i^T \boldsymbol{\beta}^{(2)}$ , the  $T$  measurements are i.i.d. Gaussian distributed; otherwise the  $T$  measurements are independently distributed as a mixture of two equally weighted Gaussian random variables. Therefore, the problem becomes a standard estimation problem for a one dimensional Gaussian mixture distributions. We propose an EM algorithm with an initialization step using moment methods to estimate the two centers of the mixture. The performance of our proposed EM algorithm can be characterized by the following lemma:

**Lemma 7.** *There exists a constant  $\eta > 0$  such that when  $\Delta/\sigma > \eta$ , by using  $R = \Theta(\text{polylog}(n))$  measurements, the proposed EM algorithm with initialization via moment methods can estimate the true two centers<sup>4</sup> of the mixture of Gaussian distributions with probability  $1 - \mathcal{O}(1/\text{poly}(n))$ .*

Details of this algorithm can be found in Appendix F.

**Consecutive Summation Check.** After the noise reduction operations, for each query vector  $\mathbf{x}_i$ ,  $i \in [P]$ , we get at most two “centers”  $\{y_{i,1}, y_{i,2}\}$  (called *denoised measurements*), which correspond to the inner product of the query vector and the parameter vectors in the noiseless case. However, we do not know the correspondence between the denoised measurements and the two parameter vectors. This means that we can have either  $(y_{i,1}, y_{i,2}) = (\mathbf{x}_i^T \boldsymbol{\beta}^{(1)}, \mathbf{x}_i^T \boldsymbol{\beta}^{(2)})$  or  $(y_{i,1}, y_{i,2}) = (\mathbf{x}_i^T \boldsymbol{\beta}^{(2)}, \mathbf{x}_i^T \boldsymbol{\beta}^{(1)})$ . Therefore, we need to use the consecutive summation check method to find the denoised measurements which are generated by the same parameter vector.

We illustrate the consecutive summation check process using a simple example in Figure 9. Assume that we have three query vectors  $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ , and two summation check queries  $\mathbf{x}_1 + \mathbf{x}_2$  and  $\mathbf{x}_2 + \mathbf{x}_3$ . Suppose that the denoised measurements that we get for  $\mathbf{x}_i$ ,  $i = 1, 2, 3$  are  $\{y_{1,1}, y_{1,2}\} = \{1, 5\}$ ,  $\{y_{2,1}, y_{2,2}\} = \{2, 4\}$ , and  $\{y_{3,1}, y_{3,2}\} = \{2, 3\}$ , and the denoised measurements for the summation check queries are  $\{y_{(1,2),1}, y_{(1,2),2}\} = \{5, 7\}$  and  $\{y_{(2,3),1}, y_{(2,3),2}\} = \{5, 6\}$ . By *matching* summations, one can easily find that the only possible case that we can observe these denoised measurements is that  $\{y_{1,1}, y_{2,2}, y_{3,1}\}$  and  $\{y_{1,2}, y_{2,1}, y_{3,2}\}$  are generated by the same parameter vector (we call them *consistent sets*), respectively, as shown in different colors in Figure 9. In our algorithm, we need to conduct consecutive summation check on all the denoised indexing and verification measurements, using the denoised summation check measurements. We also mention that the reason that we need summations of all the  $\binom{P_1+P_2}{2}$  pairs of the first  $P_1 + P_2$  query vectors is that we might have the two denoised measurements taking the same value, i.e.,  $\mathbf{x}_i^T \boldsymbol{\beta}^{(1)} = \mathbf{x}_i^T \boldsymbol{\beta}^{(2)}$ , and then we have to conduct summation check on two query vectors which are not adjacent.

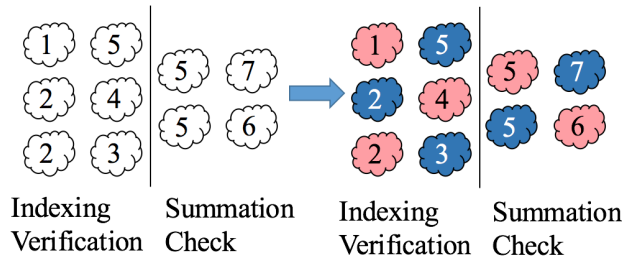


Figure 9: Consecutive summation check

**Singleton Detection.** Then, we conduct singleton detection on the consistent sets. Without loss of generality we assume  $\{y_{1,1}, \dots, y_{P_1+P_2,1}\}$  is a consistent set.<sup>5</sup> We check the first  $P_1$  denoised indexing measurements. One can easily see that the only situation where the consistent set might be a singleton is that all the non-zero denoised indexing measurements take the same value in  $\mathbb{D}$ , say  $a\Delta$ . The only possible location index  $j$  of the singleton ball satisfies the fact that  $j - 1$  has binary representation  $\{\frac{1}{a\Delta} y_{i,1}\}_{i=1}^{P_1}$ . To verify the

<sup>4</sup>Note that in our problem, the centers take quantized values, so the estimation can take the exact value of the true centers.

<sup>5</sup>Otherwise we can make them consistent by exchanging the denoised measurements.

claim that this consistent set is a singleton, we check the next  $P_2$  denoised verification measurements. If we have  $y_{i,1} = a\Delta V_{i-P_2,j}$  for all  $i = P_1 + 1, \dots, P_1 + P_2$ , then by one of the corollaries of Johnson-Lindenstrauss Lemma [30], we know that using  $P_2 = \Theta(\log(n))$  verification measurements, we can guarantee that with probability  $1 - \mathcal{O}(1/\text{poly}(n))$ , this consistent set is indeed a singleton with the singleton ball located at the  $j$ -th coordinate and taking value  $a\Delta$ .

Since the binary representation design of the queries achieves the same goal as our previous design, the other parts of our algorithm stays the same as the previous algorithm. So far, we have gathered all the ingredients for the robust Mixed-coloring algorithm and the performance can be analyzed using similar methods as the in the noiseless case.

## E Proof of Theorem 2

In this section, we analyze the performance of the robustified Mixed-Coloring algorithm and prove Theorem 2. The analysis is based on a simple application of total law of probability. We define the error event  $E_\ell$  that less than  $1 - p^*$  fraction of the  $K_\ell$  non-zero elements of the parameter vector  $\beta^{(\ell)}$  is recovered by the algorithm. We define another three events  $E_\ell^1, E_\ell^2$  as the event that there exists one incidence that the EM algorithm provides the wrong answer, and the event that there exists one incidence the verification queries (recall that the verification query vectors consist of i.i.d. Rademacher entries) misclassify a singleton as a non-singleton or wrongly identify a non-singleton as a singleton.

Then, we can analyze the error probability. First, using the same density evolution technique used in [26], which is also similar to what we have done in the proof of Theorem 1, we know that  $\mathbb{P}\{E_\ell | \bar{E}_\ell^1 \cap \bar{E}_\ell^2\} = \mathcal{O}(1/K_\ell)$ . On the other hand, by Lemma 7 and a union bound over all EM operations, we know that  $\mathbb{P}\{E_\ell^1\} = \mathcal{O}(1/\text{poly}(n))$ , and as we have mentioned in Appendix D, by one of the corollaries of Johnson-Lindenstrauss lemma and a union bound over all the singleton verification operations, we know that  $\mathbb{P}\{E_\ell^2\} = \mathcal{O}(1/\text{poly}(n))$ . Then, we know that

$$\begin{aligned} \mathbb{P}\{E_\ell\} &= \mathbb{P}\{E_\ell | \bar{E}_\ell^1 \cap \bar{E}_\ell^2\} \mathbb{P}\{\bar{E}_\ell^1 \cap \bar{E}_\ell^2\} + \mathbb{P}\{E_\ell | E_\ell^1 \cup E_\ell^2\} \mathbb{P}\{E_\ell^1 \cup E_\ell^2\} \\ &\leq \mathbb{P}\{E_\ell | \bar{E}_\ell^1 \cap \bar{E}_\ell^2\} + \mathbb{P}\{E_\ell^1 \cup E_\ell^2\} \\ &\leq \mathbb{P}\{E_\ell | \bar{E}_\ell^1 \cap \bar{E}_\ell^2\} + \mathbb{P}\{E_\ell^1\} + \mathbb{P}\{E_\ell^2\} \\ &= \mathcal{O}(1/K_\ell). \end{aligned}$$

Then, using a union bound over all the  $L$  parameter vectors, we can prove the error probability of the algorithm. Other results can be derived using the same methods as in the proof of Theorem 1.

## F Parameter Estimation for Mixtures of Gaussian Random Variables

In this section, we provide a method to estimate the parameters of a mixture of two Gaussian random variables, and give the theoretical analysis to prove Lemma 7 in the main paper. This estimation method is a combination of a moment method and the EM algorithm.

Let  $z_i$ 's be i.i.d. samples of Bernoulli( $\frac{1}{2}$ ) distribution, and  $w_i$ 's be i.i.d. samples of Gaussian distribution with mean zero and variance  $\sigma^2$ , independently of  $z_i$ 's,  $i \in [N]$ . Suppose that random variables  $y_i$ 's are generated in the following way:

$$y_i = \mu_1(1 - z_i) + \mu_2 z_i + w_i, \quad i \in [N].$$

Then, we can consider  $y_i$  as a mixture of two Gaussian random variables with means  $\mu_1$  and  $\mu_2$ , respectively. We assume that  $\sigma^2$  is known, and the parameters  $\mu_1$  and  $\mu_2$  are unknown and take value in a finite and

quantized set  $\mathbb{D} = \{k\Delta : k \in \mathbb{Z}, |k| \leq b\}$ , for some  $\Delta > 0$ . Without loss of generality, we assume that  $\mu_1 \leq \mu_2$ . (Note that we allow  $\mu_1 = \mu_2$  here.) Our goal is to get accurate estimation of  $\mu_1$  and  $\mu_2$ .

The first step is to compute the sample mean of the first  $N_1$  samples, i.e.,  $\bar{y} = \frac{1}{N_1} \sum_{i=1}^{N_1} y_i$ . Since we know that the mean of  $y_i$ 's takes value in the set  $\mathbb{D}^+ = \{\frac{k}{2}\Delta : k \in \mathbb{Z}, |k| \leq 2b\}$ , we find the element in  $\mathbb{D}^+$  which is the closest one to  $\bar{y}$  as the estimator of the mean of  $y_i$ , i.e.,  $\frac{1}{2}(\mu_1 + \mu_2)$ ,

$$\hat{\mu} = \arg \min_{\mu \in \mathbb{D}^+} |\bar{y} - \mu|.$$

We have the following result on the accuracy of the estimator  $\hat{\mu}$ .

**Lemma 8.** *There exist universal constants  $c_1$  and  $c_2$  such that for any  $\delta > 0$ , if*

$$N_1 \geq \max\{c_1 b^2, c_2 \frac{\sigma^2}{\Delta^2}\} \log(\frac{1}{\delta}), \quad (13)$$

*we have  $\hat{\mu} = \frac{1}{2}(\mu_1 + \mu_2)$  with probability at least  $1 - 6\delta$ .*

We prove Lemma 8 in Appendix F.1. In the second step, we subtract  $\hat{\mu}$  from the other  $N - N_1$  samples, and get centered random variables  $\tilde{y}_i = y_i - \hat{\mu}$ ,  $i = N_1 + 1, \dots, N$ . We assume that  $\hat{\mu}$  is the actual mean of the  $y_i$ 's, meaning that  $\hat{\mu} = \frac{1}{2}(\mu_1 + \mu_2)$ . Then, we know that if  $\mu_1 = \mu_2$ , the centered random variables  $\tilde{y}_i$ 's are i.i.d.  $\mathcal{N}(0, \sigma^2)$  distributed; otherwise  $\tilde{y}_i$ 's are i.i.d. mixtures of two Gaussian distributions

$$\tilde{y}_i \sim \begin{cases} \mathcal{N}(\theta_*, \sigma^2) & \text{with probability } \frac{1}{2} \\ \mathcal{N}(-\theta_*, \sigma^2) & \text{with probability } \frac{1}{2}, \end{cases}$$

where  $\theta_* = \frac{1}{2}(\mu_2 - \mu_1) \geq 0$ . Then, we make an initial estimation of  $\theta_*$  using  $N_2$  of the  $N - N_1$  centered random variables. Specifically, we compute

$$\theta_0 = \begin{cases} \sqrt{\frac{1}{N_2} \sum_{i=N_1+1}^{N_1+N_2} \tilde{y}_i^2 - \sigma^2} & \text{if } \frac{1}{N_2} \sum_{i=N_1+1}^{N_1+N_2} \tilde{y}_i^2 - \sigma^2 > 0 \\ 0 & \text{otherwise.} \end{cases}$$

We have the following result on  $\theta_0$ :

**Lemma 9.** *Condition on the event that  $\hat{\mu} = \frac{1}{2}(\mu_1 + \mu_2)$ . There exist universal constants  $c_3$  and  $c_4$ , such that for any  $\delta > 0$ , when*

$$N_2 \geq \max\{c_3 \frac{\sigma^2}{\Delta^2} (1 + \frac{4\sigma^2}{\Delta^2}), c_4\} \log(\frac{1}{\delta}), \quad (14)$$

*then  $\theta_0$  satisfies:*

- (1) *if  $\mu_1 = \mu_2$ ,  $\theta_0 < \frac{\Delta}{4}$  with probability at least  $1 - 2\delta$ ;*
- (2) *if  $\mu_1 \neq \mu_2$ ,  $|\theta_0 - \theta_*| < \frac{\theta_*}{4}$  with probability  $1 - 2\delta$ .*

We prove Lemma 9 in Appendix F.2. If  $\theta_0 < \frac{\Delta}{4}$ , we claim that  $\mu_1 = \mu_2$ , and give estimators  $\hat{\mu}_1 = \hat{\mu}_2 = \hat{\mu}$ . Otherwise, we run a standard EM algorithm with the remaining  $N_3 := N - (N_1 + N_2)$  samples using  $\theta_0$  as an initialization to estimate  $\theta_*$ . Here, we briefly review the procedures of standard EM algorithm for mixtures of Gaussian distributions. For  $t = 0, 1, 2, \dots$ , conduct the following two steps:

**E step:** compute the expected log-likelihood.

$$L(\theta|\theta_t) = -\frac{1}{2N_3} \sum_{i=N_1+N_2+1}^N [p(\tilde{y}_i|\theta_t)(\tilde{y}_i - \theta_t)^2 + (1 - p(\tilde{y}_i|\theta_t))(\tilde{y}_i + \theta_t)^2],$$

where

$$p(y|\theta_t) = e^{-\frac{(y-\theta_t)^2}{2\sigma^2}} \left[ e^{-\frac{(y-\theta_t)^2}{2\sigma^2}} + e^{-\frac{(y+\theta_t)^2}{2\sigma^2}} \right]^{-1}.$$

**M step:** compute

$$\theta_{t+1} = \arg \max_{\theta} L(\theta|\theta_t) = \frac{1}{N_3} \left[ 2 \sum_{i=N_1+N_2+1}^N p(\tilde{y}_i|\theta_t) \tilde{y}_i - \sum_{i=N_1+N_2+1}^N \tilde{y}_i \right].$$

We run the EM algorithm  $T$  iterations, and find the element in  $\mathbb{D}^+$  which is the closest one to  $\theta_t$  as the estimator of the mean of  $\theta_*$ , i.e.,  $\hat{\theta}_* = \arg \min_{\theta \in \mathbb{D}^+} |\theta - \theta_t|$ . Then, we output the estimation of  $\mu_1$  and  $\mu_2$  by  $\hat{\mu}_1 = \hat{\mu} - \hat{\theta}_*$  and  $\hat{\mu}_2 = \hat{\mu} + \hat{\theta}_*$ .

Here, we review the results in [14] which characterizes the performance of the EM algorithm.

**Lemma 10.** [14] Suppose that  $\mu_1 < \mu_2$ . Conditioned on the event that  $\hat{\mu} = \frac{1}{2}(\mu_1 + \mu_2)$  and the event that  $|\theta_0 - \theta_*| < \frac{\theta_*}{4}$ . Suppose that  $\eta := \frac{\theta_*}{\sigma}$  is large enough. Then, there exist universal constants  $c_5$ ,  $c_6$ , and  $c_7$ , such that when  $N_3 \geq c_5 \log(\frac{1}{\delta})$ , for any  $\delta > 0$ , we have

$$|\theta_t - \theta_*| \leq \kappa^t |\theta_0 - \theta_*| + \frac{c_6}{1 - \kappa} \theta_* \sqrt{\theta_*^2 + \sigma^2} \sqrt{\frac{1}{N_3} \log(\frac{1}{\delta})},$$

with probability at least  $1 - \delta$ , where  $\kappa \leq \exp(-c_7 \eta^2)$ .

Then, we have the direct corollary:

**Corollary 1.** Under the same condition that  $\hat{\mu} = \frac{1}{2}(\mu_1 + \mu_2)$ ,  $|\theta_0 - \theta_*| < \frac{\theta_*}{4}$ , and that  $\eta = \frac{\theta_*}{\sigma}$  is large enough as in Lemma 7, then, when

$$N_3 > \max\{c_5, \frac{16c_6^2}{(1 - \kappa)^2} b^2 (b^2 \Delta^2 + \sigma^2)\} \log(\frac{1}{\delta}), \quad (15)$$

and

$$T > \frac{\log(b)}{\log(1/\kappa)}, \quad (16)$$

we have  $\hat{\theta}_* = \theta_*$  with probability at least  $1 - \delta$ , for any  $\delta > 0$ .

We prove Corollary 1 in Appendix F.3. We have the following theorem to characterize the performance of the proposed estimation algorithm.

**Theorem 3.** If  $N_1$ ,  $N_2$ ,  $N_3$ , and  $T$  satisfy (13), (14), (15), and (16), respectively, and  $\frac{\Delta}{\sigma}$  is large enough, then the proposed estimation algorithm outputs correct estimations  $\hat{\mu}_1 = \mu_1$  and  $\hat{\mu}_2 = \mu_2$  with probability at least  $1 - 9\delta$ , for any  $\delta > 0$ .

*Proof.* Let  $A_1$  and  $A_2$  be the events that  $\hat{\mu} = \frac{1}{2}(\mu_1 + \mu_2)$  and that  $\hat{\theta}_* = \theta_*$ , respectively, and  $A$  be the event that  $\hat{\mu}_1 = \mu_1$  and  $\hat{\mu}_2 = \mu_2$ . Then, by Lemma 8, we know that  $\mathbb{P}\{A_1\} \geq 1 - 6\delta$ .

If  $\mu_1 = \mu_2$ , by Lemma 9, we know that  $\mathbb{P}\{A|A_1\} \geq 1 - 2\delta$ . Then  $\mathbb{P}\{A\} \geq \mathbb{P}\{A|A_1\}\mathbb{P}\{A_1\} \geq 1 - 8\delta$ . If  $\mu_1 < \mu_2$ , by Lemma 9, we know that  $\mathbb{P}\{A_2|A_1\} \geq 1 - 2\delta$ , and by Corollary 1, we know that  $\mathbb{P}\{A_3|A_2, A_1\} \geq 1 - \delta$ . Then,  $\mathbb{P}\{A\} \geq \mathbb{P}\{A_1\}\mathbb{P}\{A_2|A_1\}\mathbb{P}\{A_3|A_2, A_1\} \geq 1 - 9\delta$ .  $\square$

Then, we can derive Lemma 7 in the main paper by setting  $\delta = 1/\text{poly}(n)$  and  $R = N = N_1 + N_2 + N_3$ .

## F.1 Proof of Lemma 8

First, we can see that to get an accurate estimation, it suffices to have  $|\bar{y} - \frac{1}{2}(\mu_1 + \mu_2)| < \frac{\Delta}{4}$ . Let  $N_{11} = \sum_{i=1}^{N_1} 1 - z_i$ , and  $N_{12} = \sum_{i=1}^{N_1} z_i$ . We have

$$\bar{y} = \frac{N_{11}}{N_1} \mu_1 + \frac{N_{12}}{N_1} \mu_2 + \frac{1}{N_1} \sum_{i=1}^{N_1} w_i.$$



By Hoeffding's inequality, we have

$$\mathbb{P} \left\{ \left| \frac{N_{11}}{N_1} \mu_1 - \frac{\mu_1}{2} \right| < \frac{\Delta}{12} \right\} \geq 1 - 2 \exp(-\frac{\Delta^2 N_1}{72 \mu_1^2}) \geq 1 - 2 \exp(-\frac{N_1}{72 B^2}), \quad (17)$$

and similarly

$$\mathbb{P} \left\{ \left| \frac{N_{12}}{N_1} \mu_2 - \frac{\mu_2}{2} \right| < \frac{\Delta}{12} \right\} \geq 1 - 2 \exp(-\frac{\Delta^2 N_1}{72 \mu_2^2}) \geq 1 - 2 \exp(-\frac{N_1}{72 b^2}). \quad (18)$$

By Chernoff's inequality, we have

$$\mathbb{P} \left\{ \left| \frac{1}{N_1} \sum_{i=1}^{N_1} w_i \right| < \frac{\Delta}{12} \right\} \geq 1 - 2 \exp(-\frac{N_1 \Delta^2}{288 \sigma^2}). \quad (19)$$

By triangle inequality and union bound, we get

$$\mathbb{P} \left\{ \left| \bar{y} - \frac{1}{2}(\mu_1 + \mu_2) \right| < \frac{\Delta}{4} \right\} \geq 1 - 4 \exp(-\frac{N_1}{72 b^2}) - 2 \exp(-\frac{N_1 \Delta^2}{288 \sigma^2}),$$

which completes the proof.

## F.2 Proof of Lemma 9

Let  $A_1$  be the event that  $\hat{\mu} = \frac{1}{2}(\mu_1 + \mu_2)$ . In this lemma, all the probabilities are conditioned on the event  $A_1$ .

First, consider the case when  $\mu_1 = \mu_2$ , i.e.,  $\theta_* = 0$ . Let  $\tilde{y} := \frac{1}{\sigma^2} \sum_{i=N_1+1}^{N_1+N_2} \tilde{y}_i^2$ . Then, we know that  $\tilde{y}$  is  $\chi^2$  distributed with  $N_2$  degrees of freedom. By the concentration result of  $\chi^2$  distribution, we have for any  $\epsilon > 0$ ,

$$\mathbb{P} \left\{ \left| \frac{1}{N_2} \tilde{y} - 1 \right| \geq \epsilon | A_1 \right\} \leq 2 \exp(-\frac{N_2}{8} \min\{1, \epsilon^2\}).$$

Then, we have

$$\mathbb{P} \left\{ \theta_0 < \frac{\Delta}{4} | A_1 \right\} \geq \mathbb{P} \left\{ \left| \frac{\tilde{y}^2}{N_2} - 1 \right| < \frac{\Delta^2}{16 \sigma^2} | A_1 \right\} \geq 1 - 2 \exp(-\frac{N_2}{8} \min\{1, \frac{\Delta^2}{16 \sigma^2}\}),$$

which implies that if

$$N_2 \geq 8 \max\{1, 16 \frac{\sigma^2}{\Delta^2}\} \log(\frac{1}{\delta}), \quad (20)$$

conditioned on  $A_1$ , the probability that  $\theta_0 < \frac{\Delta}{4}$  is at least  $1 - 2\delta$ .

Then, we consider the case when  $\mu_1 \neq \mu_2$ . In this case, we have  $\theta_* \geq \frac{\Delta}{2}$ , and we study the probability that  $|\theta_0 - \theta_*| \leq \frac{\theta_*}{4}$ . We still define  $\tilde{y} := \frac{1}{\sigma^2} \sum_{i=N_1+1}^{N_1+N_2} \tilde{y}_i^2$ . We can see that  $\tilde{y}$  has noncentral  $\chi^2$  distribution with  $N_2$  degrees of freedom and noncentrality parameter  $\nu = N_2 \frac{\theta_*^2}{\sigma^2}$ . According to the results of concentrations of non-central  $\chi^2$  distribution, we have for all  $\epsilon > 0$ ,

$$\mathbb{P} \left\{ \tilde{y} \geq (N_2 + \nu) + 2\sqrt{(N_2 + \nu)\epsilon} + 2\epsilon | A_1 \right\} \leq \exp(-\epsilon), \quad (21)$$

$$\mathbb{P} \left\{ \tilde{y} \leq (N_2 + \nu) - 2\sqrt{(N_2 + 2\nu)\epsilon} | A_1 \right\} \leq \exp(-\epsilon). \quad (22)$$

We analyze the probability that  $\frac{\theta_0}{\theta_*} < \frac{5}{4}$ . We substitute  $\tilde{y}$  and  $\nu$  in (21) with  $N_2(\frac{\theta_0^2}{\sigma^2} + 1)$  and  $N_2 \frac{\theta_*^2}{\sigma^2}$ , respectively. By some rearrangements, we get

$$\mathbb{P} \left\{ \frac{\theta_0^2}{\theta_*^2} \geq 1 + 2 \frac{\sigma}{\theta_*^2} \sqrt{\frac{(\theta_*^2 + \sigma^2)\epsilon}{N_2}} + \frac{2\sigma^2\epsilon}{\theta_*^2 N_2} | A_1 \right\} \leq \exp(-\epsilon).$$

Then, we know that if we can guarantee that  $\frac{\sigma}{\theta_*^2} \sqrt{\frac{(\theta_*^2 + \sigma^2)\epsilon}{N_2}} \leq \frac{9}{64}$  and  $\frac{\sigma^2 \epsilon}{\theta_*^2 N_2} \leq \frac{9}{64}$ . Considering the fact that  $\theta_* \geq \frac{\Delta}{2}$ , we know that there exists universal constants  $c_3$  such that if  $N_2$  satisfies

$$N_2 \geq c_3 \frac{\sigma^2}{\Delta^2} \left(1 + \frac{4\sigma^2}{\Delta^2}\right) \log\left(\frac{1}{\delta}\right), \quad (23)$$

then the probability that  $\frac{\theta_0}{\theta_*} < \frac{5}{4}$  conditioned on  $A_1$  is at least  $1 - \delta$ . Similarly, using (22), we know that when (23) is satisfied, we can guarantee that  $\frac{\theta_0}{\theta_*} > \frac{3}{4}$  with probability at least  $1 - \delta$ . We can complete the proof by union bound.

### F.3 Proof of Corollary 1

To guarantee that  $\hat{\theta}_* = \theta_*$ , we need  $|\theta_T - \theta_*| < \frac{\Delta}{2}$ . By Lemma 9, it suffices to have  $\kappa^T |\theta_0 - \theta_*| < \frac{\Delta}{4}$  and  $\frac{c_6}{1-\kappa} \theta_* \sqrt{\theta_*^2 + \sigma^2} \sqrt{\frac{1}{N_3} \log\left(\frac{1}{\delta}\right)} < \frac{\Delta}{4}$ . Taking the conditions that  $|\theta_0 - \theta_*| < \frac{\theta_*}{4}$  and  $\theta_* < b\Delta$ , we know that it is sufficient to have  $T > \frac{\log(b)}{\log(1/\kappa)}$  and  $N_3 > \frac{16c_6^2}{(1-\kappa)^2} b^2 (b^2 \Delta^2 + \sigma^2) \log\left(\frac{1}{\delta}\right)$ .